Earth Sci Inform
DOI 10.1007/s12145-014-0185-z

RESEARCH ARTICLE

Intelligent polar cyberinfrastructure: enabling semantic search in geospatial metadata catalogue to support polar data discovery

Wenwen Li · Vidit Bhatia · Kai Cao

Received: 25 March 2014 / Accepted: 7 October 2014

Abstract Polar regions have garnered substantial research attention in recent years because they are key drivers of the Earth's climate, a source of rich mineral resources, and the home of a variety of marine life. Nevertheless, global warming over the past century is pushing the polar systems towards a tipping point: the systems are at high-risk from melting snow and sea ice covers, permafrost thawing, and acidification of the Arctic oceans. To increase understanding of the polar environment, the National Science Foundation established a Polar Cyberinfrastructure (CI) program, aimed at utilizing advanced software architecture to support polar data analysis and decision-making. At the center of this Polar CI research are data resources and data discovery components that facilitate the search and retrieval of polar data. This paper reports our development of a semantic search tool that supports the intelligent discovery of polar datasets. This tool is built on latent semantic analysis techniques, which improves search performance by identifying hidden semantic associations between terminologies used in the various datasets' metadata. The software tool is implemented using an object-oriented design pattern and has been successfully integrated into a popular open source metadata catalog as a new semantic search support. A semantic matrix is maintained persistently within the catalogue to store the semantic associations. A dynamic update mechanism was also developed to allow automated update of semantics once more metadata are loaded into or removed from the catalog. We explored the effects of rank reduction to the effectiveness of this semantic search module and demonstrated its better performance than the traditional search techniques.

Keywords Polar · Arctic · Antarctic · Spatial data infrastructure · Metadata · Catalog · Discovery

Introduction

The Polar regions are major sources of mineral resources, and they are also home of a variety of marine life, thus are of great importance to our planet. As key drivers of the Earth's climate, environmental changes in Polar Regions signal global climate change. They drive the environment at lower latitudes, through impacts on atmospheric circulation of greenhouse gases (Marshall et al. 2013), changes in river runoff (Overpeck 1997; Gosling et al. 2011), as well as effects on thermohaline circulation across oceans (Stouffer et al. 2006; Goelzer et al. 2011). Unfortunately, global warming over the past century is pushing the polar systems towards a tipping point: the systems are at high-risk from melting snow and sea ice covers, permafrost thawing, and acidification of the Arctic oceans (Scudellari 2013). Studies predict that the North Pole may expect an ice-free summer by 2040 (Holland et al. 2006; Cochran et al. 2013). A consequence would be sea level rise, affecting more than 600 million people living in the low-lying regions (Nicholls et al. 2011). The melting of snow and ice cover in the North Pole will also endanger the habitat of ice-dependent wildlife. Warming effects could lead to the release of larger amounts of carbon dioxide and methane, from the thawing permafrost to the atmosphere, further expediting global warming process (Zimov et al. 2006). Therefore, there is a pressing need for new spatial data

Communicated by: H. A. Babaie

Published in the Special Issue "*Semantic e-Sciences*" with Guest Editors Dr. Xiaogang Ma, Dr. Peter Fox, Dr. Thomas Narock and Dr. Brian Wilson

W. Li (✉) · V. Bhatia
GeoDa Center for Geospatial Analysis and Computation, School of Geographical Sciences and Urban Planning, Arizona State University Tempe, Phoenix, AZ 85004, USA
e-mail: wenwen@asu.edu

K. Cao
Department of Geography, National University of Singapore, 117570 Singapore, Singapore

Published online: 28 October 2014

infrastructure to archive historical Earth observation data, as well as a method for researchers and decision makers access the data, so that they may understand polar changes and ultimately protect polar ecosystem and our planet as a whole.

Cyberinfrastructure (CI), which is an integration of high performance hardware, software, and network, represents the trend of next generation software infrastructure and is a promising technique to address the data challenges faced by the polar science community (Li et al. 2011a, b). A number of Polar Cyberinfrastructure projects have been launched in the past few years. AOOS (Alaska Ocean Observing System) Workspace provides an online gateway to allow data uploading, sharing, storing and organizing between members of the biological and physical oceanography communities. For the Antarctic region, the Lamont-Doherty Earth Observatory provides access to geoscience data through the Antarctic and Southern Ocean Data Portal. The Advanced Cooperative Arctic Data and Information Service (ACADIS) provides a web portal for archiving, browsing, preserving and accessing the data acquired by the Arctic Observation Network (AON). The Arctic Spatial Data Infrastructure (SDI) project aims at providing access to spatially related reliable information over the Arctic to facilitate monitoring and decision-making (Skedsmo et al. 2011). The National Snow and Ice Data Center (NSIDC) also provides an online catalog and a search tool to access all NSIDC datasets related to snow and ice.

Core components of the above CI portals are a database and an associated front-end search tool (Li et al. 2008a, b). The geospatial database, or "geospatial data clearinghouse", stores all available data records. The front-end search tool enables effective discovery of the most appropriate datasets. Lucene, a Java-based full-text indexing and searching technique, is usually used in these portals to support keyword-based searches because it is open-source, introduces efficient indexing mechanisms, and achieves relatively good search performance. Lucene's search process is based on matching keywords within the metadata and the search. Ideally, the metadata contains the exact same keywords as an end user's query, and the dataset would be identified as relevant. However, natural language is so flexible that many semantically related keywords are spelled differently– the synonym issue. Additionally, the same keyword may have different meanings when used in a metadata and in a search request—the polyseme issue (Li et al. 2008a). These issues are common in the context of polar science research. In these situations Lucene search, unfortunately, may not be able to establish latent semantic links between queries and datasets. Therefore, it is necessary to create smarter and more effective search techniques.

This paper introduces our use of latent semantic analysis in a semantic search tool for intelligent polar data discovery. We implement our algorithm in a popular open-source metadata catalog—Geonetwork, which is the core software platform of GEOSS (Global Earth Observation System of Systems) clearinghouse (Liu et al. 2011) and many other geospatial clearinghouse solutions, such as OneGeology, Dutch National Georegistry, etc. Section 2 introduces existing efforts in the literature to support data search. Section 3 describes the key algorithm, the search workflow, and the service-oriented integration in the proposed search framework. Section 4 demonstrates the Graphic User Interface (GUI) of the search tool as an enhanced version of Geonetwork. Section 5 compares the performance of Lucene and the proposed semantic search algorithm in terms of both recall and precision. Section 6 concludes our work and discusses future research directions.

Existing efforts in effective discovery of distributed data resources

The semantic-based search technique can be used to link a search request with the most relevant dataset in a data retrieval process. Compared to other widely used keyword matching techniques, semantic search is an improvement because it can identify keywords that are exact matches and those that are close in meaning (Jones et al. 2004). This search strategy can be categorized into two classes: ontology-aided semantic search and smart search based on knowledge mining. Below we review each of these categories.

Top-down ontology-based data search

The ontology-based approach is a top-down approach: relationships among concepts are pre-defined by domain experts and encoded in a machine-understandable format. When a search request is sent, a semantic search engine tries to understand the contextual meaning of search keywords aided by the ontology in the hope of generating more relevant results. One commonly adopted semantic search method is ontology-based query expansion. For instance, Li et al. (2011a) presents a semantic search testbed that utilizes a domain ontology SWEET (Semantic Web for Earth and Environmental Terminology) and semantic reasoning for identifying semantically related datasets. Superclasses, subclasses, and relevant terms in the same or different realms are defined in SWEET. For instance, "Precipitation" is a superclass of "Rainfall"; "Flooding" as a process is linked with "Stream" in the earth realm facet. "Permafrost" and "cryotic soil" are defined as relevant terms. These ontological relationships are represented in triples, composed by "Subject", "Predicate" and "Object", and are loaded in a triple store. When conducting ontological reasoning, a reasoner will convert a search statement into a SPARQL (Semantic Web Query Language) query. This query will return keywords that can be more specialized, more general, or otherwise related to the original ones. By selecting a relevant keyword, $\{i\}$, recommended by the search tool, the users can view (1) metadata records containing $\{i\}$ as search results; and (2) another set of relevant keywords $\{j\}$ that are similar in meaning or relevant to $\{i\}$ through further reasoning. Through this semantic navigation and keyword refining process, the linkage between a user query and the best matching dataset can be identified. Similar works of this kind that can also be called multi-faceted or view-based searches include: Bernard et al. (2003), Hyvönen et al. (2004), Ramachandran et al. (2006), Budak Arpinar et al. (2006), Beran (2007), Aguilar-Lopez et al. (2009), Xiong et al. (2009), Wang (2013), Bhogal et al. (2007), Castells et al. 2007, and Fernández et al. (2011).

In addition to the use of ontology for query expansion, research has been conducted about the use of ontology and reasoning to interpret natural languages queries (Lopez et al. 2005; Cimiano et al. 2007; Tran et al. 2007). These approaches fall in the category of ontology-based natural language processing.

Success of the ontological approach for semantic search is limited by the performance of the search, which is heavily dependent on the quality of the ontology (Harvey et al. 1999). This approach assumes that the semantic relationships in the data can be captured in advance. In reality, different people have different perspectives of the relationships on which the ontology must be built. It is extremely difficult to build a complete knowledge-base that can serve various perspectives and purposes. Therefore, search performance is significantly hindered when a good ontology is not available.

Bottom-up data mining-based semantic search

To address the limitation in the top-down ontology-aided semantic search approach, researchers have started to exploit the use of machine learning techniques for automatic discovery of semantic relationships. This method is called bottom-up data mining. Janowicz (2012) discussed the use of 'semantic signature' acquired from a data mining approach, Latent Dirichlet Allocation (LDA), to compare the semantic similarity between spatial features. These similarity values act as important factors to link semantically related terms in an ontology, which can be eventually used to assist the data discovery process (Celikyilmaz et al. 2010; Christidis et al. 2012). Daniel and Wood (1999), Steinbach et al. (2000), and Dhillon et al. (2001) use clustering techniques to classify large collections of text documents to improve the performance of information retrieval systems through the a of document classification taxonomy. Li et al. (2012) proposed the use of Latent Semantic Indexing (LSI; Deerwester et al. 1990; Dumais 2004) to support semantic search of geospatial data. LSI is based on the principle that words or terms appearing in the same context tend to be semantically related. Through measuring the co-existence of terms in a collection of metadata documents, known as a corpus, this method has the ability to link words with similar meanings, thereby better facilitating the information retrieval process. Similar research also includes Alhabashneh et al. (2011), and Chen et al. (2013). A detailed example of LSI is given in Section 3.2.

Challenges in semantic search

Recall and precision are important factors for search performance evaluation. Recall calculates the ratio between number of relevant results found by a search tool and the number of all relevant results in a database. The more relevant results found, the higher the recall is. Precision computes the portion of results that are actually relevant to a search from all results found by a search tool. The higher the search precision is, the fewer noisy results are included in the result.

Semantic search tools have the ability to find documents that contain not only the exact search keywords, but also words with semantically related, similar meanings. Therefore, keyword search recall rates can be greatly improved by employing a semantic search tool. However, the recall rate is also acknowledged to be limited by the scope and variation of the semantics used to support the search (Mangold 2007). At the same time, since more results are identified as relevant, the precision rate can be reduced. This tradeoff is identified in existing research (Hjørland 2010; Li et al. 2012). To overcome it, we propose the use of latent semantic indexing and a new ranking mechanism to improve search performance in terms of both recall and precision. We have successfully integrated this semantic search tool into a popular metadata catalog software, Geonetwork, to benefit the larger GIS community. The next section describes the system design and the search algorithm in detail.

Methodology

A service-oriented software framework for metadata management and discovery

Geonetwork is a powerful tool that provides metadata cataloguing, indexing, and searching functionality for geospatial resource management. As an open source catalog application, Geonetwork has been widely used in numerous national and international spatial data infrastructure projects (http://geonetwork-opensource.org). Because of its popularity, our goal in this paper, besides the development of a new semantic search tool, is to provide ways to seamlessly integrate this new search functionality into Geonetwork. This extension can improve data search in Geonetwork; at the same time, it can also make our tool more broadly available to the geospatial science community. In this section, we first introduce Geonetwork's system architecture and then introduce how we extend its existing framework to enable semantic search.

Figure 1 displays the design of Geonetwork. This software framework is built upon a Service Oriented Architecture (SOA; MacKenzie et al. 2006) and is compliant with OGC geospatial portal reference architecture (Rose 2004). Different from other service-oriented applications, i.e. Li et al. (2011b, 2013), in a distributed web environment, this SOA concept is tied to software design patterns and provides a scalable paradigm for managing large enterprise software systems. All functionality that a metadata catalog provides is implemented through services, by which the user interface and the business logic are decoupled in the software system. The user interface in this metadata catalog, the *View* part (orange boxes in Fig. 1), is responsible for displaying request responses. The business logic module, which can also be considered a software *Model* (grey boxes in Fig. 1), is responsible for communicating with database or other services to process the request. Another module, the *Controller* (pink boxes in Fig. 1), coordinates data transfer between the view and the model, controls the model's state, as well as the model's presentation in the view. This Model-View-Controller (MVC) and service-oriented software design pattern in this metadata catalog make it fully extensible and scalable.

The Geonetwork metadata catalog uses Jeeves (Java Easy Engine for Very Effective Systems) as the Java engine. It allows the separation of view layer and model layer and uses a dispatcher to control the communication between them. Given a user request, the processing flow contains:

(1) Jeeves servlet passes all incoming requests from end users to Jeeves engine;
(2) Jeeves engine extracts service name that follows certain pattern in the URL request;
(3) Jeeves engine then sends this information to the service manager;
(4) Service manager looks up the mapping table and identifies the service information, i.e., core java classes that provides a search service or a log in service;
(5) This service information is then passed to the dispatcher;
(6) The dispatcher is responsible for invoking a specific service class, i.e. the search service. This is how a service is executed;
(7) The service class communicates with backend business logic classes to fulfill a user request;
(8) The response will be encapsulated into an XML document and returned back to the service module and then to the dispatcher;
(9) The next step is to display the results in the GUI. To accomplish this task, the dispatcher calls a GUIService. The GUIService loads the class OutputPage to translate XML response into HTML webpages using a predefined XSL (eXtensible Stylesheet Language).

The proposed semantic searcher will follow the exact request/response flow in Geonetwork with the following extensions. For the view part (orange boxes), new XSLs are defined to wrap a request and display the search result. For the backend service part (grey box originating an XML request, or arrow 7), the new semantic searcher needs to be enabled to

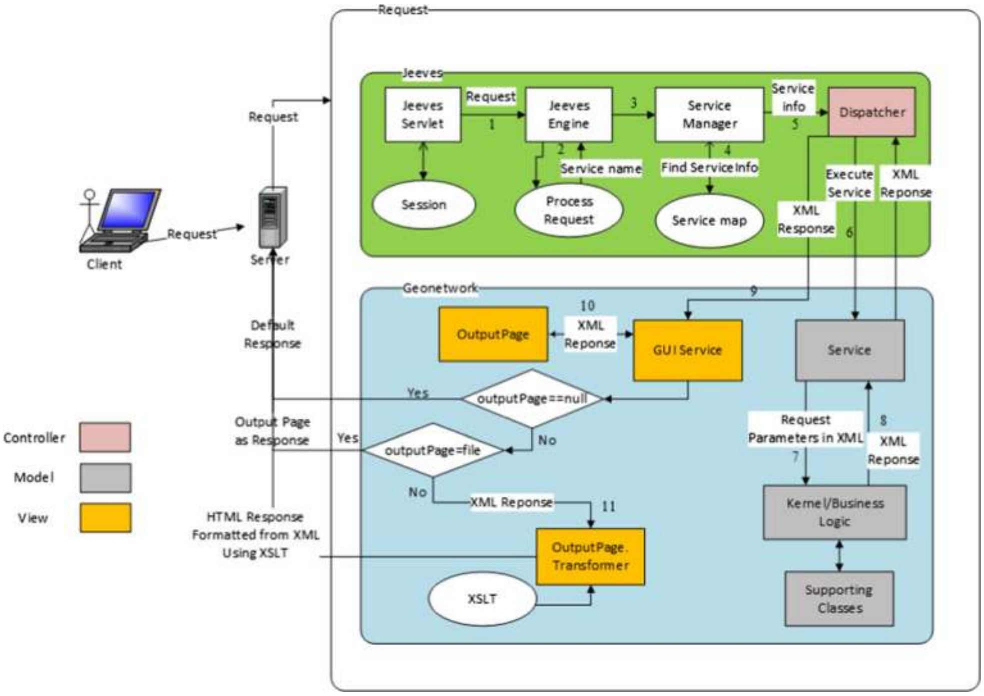


Fig. 1 Architecture of a service-oriented design of metadata catalogue

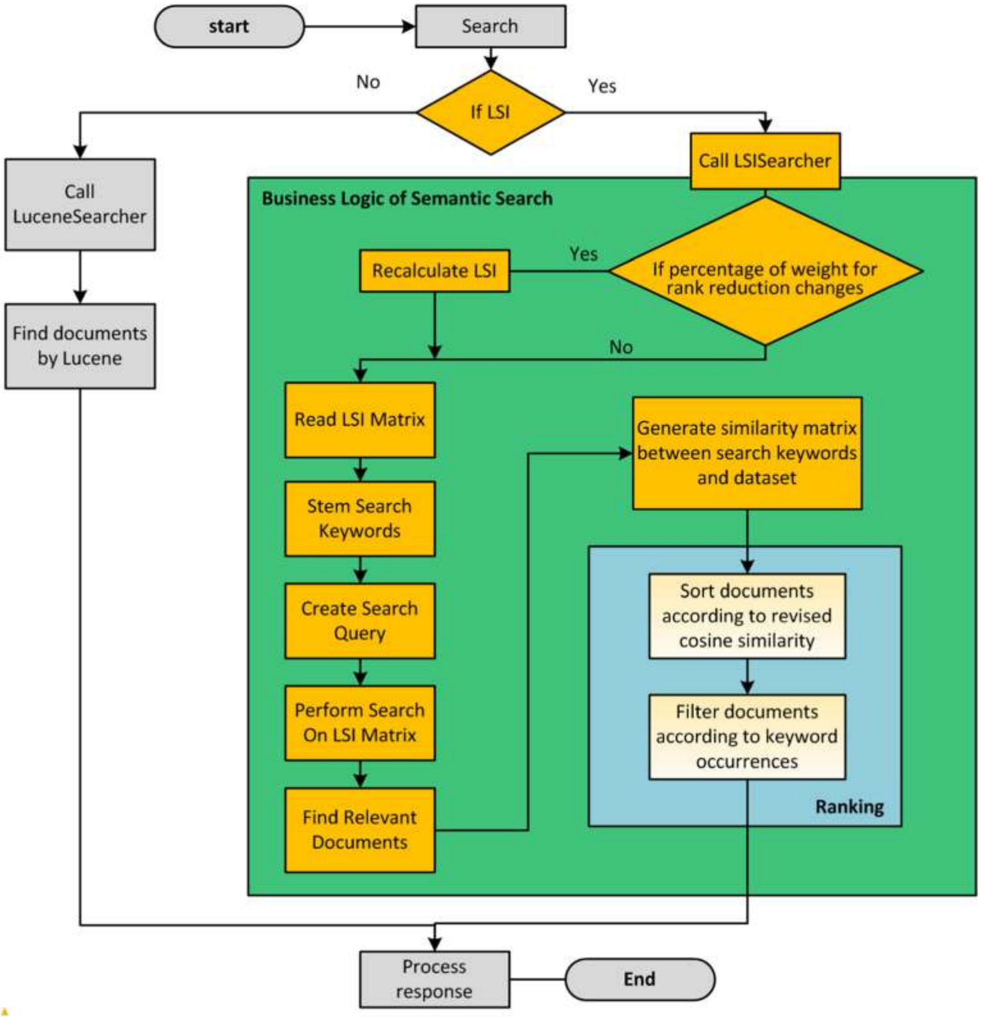
fulfill the semantic search request. The next section describes how the business logic module (grey box origin of arrow 8, the XML response) of the semantic search tool works and how it is integrated into this SOA framework.

Business logic of semantic searcher

When a search request is initiated, a search service will be invoked. This search service decides whether semantic search is supported. If not, the service will redirect this request to a LuceneSearcher (left branch in Fig. 2), which uses a full-text indexing technique and is currently supported by Geonetwork. If the semantic search is enabled, a new business logic module named LSISearcher will be invoked. This LSISearcher handles all essential steps for a search including semantic indexing, searching, and ranking of relevant documents. The LSI search builds upon and extends full-text indexing to discover the latent semantic relationships between texts. Therefore, full-text indexing is the essential first step for creating the semantic matrix to support semantic search. A regular full-text search process is first to generate a "*count matrix*", which extracts all keywords in all documents in a database, and counts the frequency of occurrence, *n*, of each keyword in each document. The document vector and the keyword vector form a matrix, and each cell stores the frequency *n*. When extracting the keywords from the document, a *stemming* process is needed to remove the inflected part of a keyword and get its root. Using stemming, words such as "act", "acts" and "acting" will only be recorded as one entry in the count matrix because they have the same root "act". Stemming ensures high recall rate by avoiding mismatches.

After a count matrix is established, a TF-IDF (Term Frequency—Inverse Document Frequency) is computed on top of the count matrix to weigh the importance of a keyword in a document. The basic principle is that words having high frequency of occurrences across all documents in a corpus are less informative than those not frequently occurring (IDF part). A word appearing many times in a document weighs more than those appearing less often (TF). The multiplication of TF value and IDF value gives an overall importance of a word. This updated matrix is called TF-IDF matrix.

Fig. 2 Business logic of the semantic search tool – LSISearcher


start
Search
No
If LSI
Yes
Call LSISearcher
Call LuceneSearcher
Find documents by Lucene
Business Logic of Semantic Search
Recalculate LSI
Yes
If percentage of weight for rank reduction changes
No
Read LSI Matrix
Stem Search Keywords
Create Search Query
Perform Search On LSI Matrix
Find Relevant Documents
Generate similarity matrix between search keywords and dataset
Sort documents according to revised cosine similarity
Filter documents according to keyword occurrences
Ranking
Process response
End

A full text search is often performed on the TF-IDF matrix instead of a raw count matrix. By computing the cosine similarity between a search vector and a document vector, a similarity value can be obtained and this value is used to rank the relevance of a result to a search request. Different from full text search, LSI further analyzes the TF-IDF matrix to discover unknown semantic relationships between words. For instance, in a TF-IDF matrix, if a keyword does not appear in a document, the cell value referring to that particular keyword and the document is 0, in both count matrix and the TF-IDF matrix. However, keywords (*sk*) with similar, semantically related meanings may appear in the document. If a linkage between a search keyword and *sk* can be found, the document will be identified as relevant in the search procedure.

LSI is capable of making this semantic linkage. On top of a TF-IDF matrix, LSI adopts a mathematical technique—Singular Value Decomposition (SVD) to decompose the matrix into the multiplication of an orthonormal term-concept matrix, a diagonal matrix and another orthonormal concept-document matrix. The values in the diagonal matrix decrease as the row/column index increases. The LSI modification to the TF-IDF matrix is through “rank reduction”. In this technique, we only retain the largest *k* values in the diagonal matrix and set the rest to be 0. Then the three matrices will be multiplied back together to create a new matrix, called LSI matrix in this paper. This modification has the ability of retaining the semantic related information and removing the noisy data in the original document. Technically, a latent semantic linkage is represented by a modified cell value in the new LSI matrix. For a document *d* containing words with similar meanings, but not the exact appearance of a keyword *sk*, the cell value showing the weight of *sk* in *d* increases from 0 to a positive weight value. For a document *d* containing *sk,* but having some theme less relevant to the keyword, the cell value indicating the weights of *sk* in *d* in the LSI matrix will be reduced from its original value. This way, latent semantics can be captured and more relevant documents can be identified.

The following is a small example to demonstrate the use of LSI. Suppose a data collection contains titles of eight documents and two disjoint topics. In the list below, c1-c4 refer to geospatial semantic search and m1-m4 refer to hydrological law. The dimensions of the term-by-document matrix A are 7*8 (Table 1). Instead, the count matrix of TF-IDF matrix is used for simplicity. Seven terms occurred at least twice in the eight documents. Therefore, each row is the keyword vector and each column is the document vector. After an SVD of the matrix A is performed, the diagram matrix S has decreasing rank values {2.74, 2.37, 1.65, 1.24, 1, 0.80, 0} at [][], where≤ 7. When using the most two important dimensions of S to construct the semantic matrix, we obtain A , as illustrated in Table 2. Notice, the keyword “hydrology” does not appear in document m4, therefore the corresponding cell [7][8] is assigned 0. However, because m3 contains terms “environment” and “hydrology”, the keyword hydrology is determined relevant to “environment”. Because m4 contains “environment”, the keyword hydrology’s cell value in m4’s column ([7][8]) has been replaced to 0.67 although this word never exists in m4. As a comparison, the term “environment” that appears in c4, and is assigned to 1 in A , is now replaced by 0.4591 in A reflecting its unimportance in characterizing the document as related to “environment” based on the latent semantic analysis.

Table 1 Original term-by-document count matrix

Matrix A	C1	C2	C3	C4	M1	M2	M3	M4
Geo	1	1	0	0	0	0	0	0
spatial	1	1	1	1	0	0	0	0
semantic	0	1	1	0	0	0	0	0
search	0	0	1	1	0	0	0	0
Environment(al)	0	0	0	1	0	0	1	1
law	0	0	0	0	1	1	0	1
hydrology	0	0	0	0	1	1	1	0

The matrix reconstruction also reveals relationships between keywords. In the original matrix A , “geo” and “search,” and “geo” and “law” never co-appear in any document. Thus, intuitively, they do not have much association. A correlation analysis using Spearman correlation analysis (Spearman 1904) shows that the correlation between “geo” and “search” is −0.33 and the value is −0.45 between “geo” and “law.” However, in the reconstructed matrix, the underlying associations are uncovered, both values have greatly changed. The correlation between “geo” and “search” increases to 0.9961 (almost the upper limit 1) and the correlation between “geo” and “law” decreases to −0.9655 (almost the lower limit −1). This is because both “geo” and “search” frequently occur in the same context as “spatial”, and the LSI analysis has uncovered this indirect relationship. Therefore, even though they do not both occur in the same document, they occur in the contexts of relatedness. For “geo” and “law,” no hidden association can be found in the context, so they are ranked very low in the correlation analysis.

c1: The geo-spatial Web: how geo-browsers, social software and the Web 2.0 are shaping the network society.
c2: Geo-spatial semantics: capture meanings of spatial information
c3: A semantic search engine for spatial Web portals
c4: Google’s spatial search tools in the Marine Environment - Decision Support
m1: Darcy’s law on hydrology
m2: Hydrology and Water Law - Bridging the Gap
m3: Hydrology: an environmental approach
m4: Environmental law: Hazardous wastes and substances

Table 2 Modified term-by-document matrix by LSI

Matrix Â	C1	C2	C3	C4	M1	M2	M3	M4
Geo	0.38	0.53	0.53	0.42	−0.11	−0.11	−0.02	−0.02
spatial	0.79	1.08	1.10	0.93	−0.07	−0.07	0.1	0.1
semantic	0.45	0.62	0.62	0.50	−0.11	−0.11	−0.00	−0.00
search	0.40	0.55	0.57	0.51	0.04	0.04	0.11	0.11
Environment(al)	0.17	0.22	0.28	0.46	0.55	0.55	0.51	0.51
law	−0.09	−0.14	−0.07	0.26	0.80	0.80	0.67	0.67
hydrology	−0.09	−0.14	−0.07	0.256	0.80	0.80	0.67	0.67

Using this technique, the LSIsearcher will execute the workflow described in Fig. 2 (yellow boxes). First, the LSIsearcher determines if the percentage of weights for rank reduction has changed in the GUI. This is a new feature made to aid the semantic search. The total weight is the sum of all values in the diagonal matrix. In rank reduction, by selecting different percentages of total weight, LSI modification will catch different levels of latent semantics in the data. For example, a rank preservation rate at 100 % means all elements in the diagonal matrix are preserved and the modified LSI matrix will be the same as the TF-IDF matrix because there is no tuning of the data. 80 % means the diagonal matrix will keep the values in a sub-matrix starting from upper left and their sum is about 80 % of the total weights. The rest of values in the diagonal matrix will be removed. This way, LSI modification will generate a new LSI matrix which shows the latent relationships between keywords and documents. If a user changes this percentage value, termed "rank preservation rate", the LSI matrix needs to be recomputed (details of this computation are discussed in section 3.3). If not, the LSIsearcher reads in the LSI matrix stored in the database, processes the search keywords to get their stems, generates the query vector and then performs the search on the LSI matrix by measuring the cosine similarity between each search vector X and each vector representing a metadata document Y.

Cosine similarity is a commonly used technique in a search engine to rank the results according to their similarity to the query. The cosine similarity in this work is different from that used in Lucene (Singhal 2001). We adopted a revised cosine similarity listed below:

$$sim(X,Y)=\frac{\sum_{i=1}^{n} X_i * Y_i}{\sqrt{\sum_{i=1}^{n}(X_i - Y_i)^2}} \tag{1}$$

where n is the total number of unique keywords in a database, and X_i and Y_i are weights of these keywords in a query vector and a document vector. Y_i is a revised weight in the LSI matrix. This similarity measure is not only able to measure the similarity based on the angle between the two vectors (query vector and document vector) from the regular cosine similarity, but it can also detect the similarity based on the distance between two vectors. Earlier experiments by Li et al. (2012) show that this measure works better in similarity ranking than the commonly used cosine similarity measure.

After the similarity ranking, a mechanism is needed to cut off the results that have a small similarity value. This is challenging because if the cut off similarity value is too high, the recall rate will be affected negatively. Whereas, a lower cutoff value reduce the precision of the results. To solve this problem, we introduce a new filter to remove the not closely relevant results. The filter will check the LSI cell values in a document for the given search keywords; if any of them has value of 0, the keyword neither exists in the metadata document nor is semantically related to it, then the metadata document will be disregarded. Using this ranking strategy, the recall can be guaranteed and the precision can be greatly improved. Experimental results in Section 4 demonstrate this improvement.

Creation and update of LSI matrix

The semantic search process relies on a LSI matrix that is precomputed to make the search more efficient. Note that a document used for indexing by the LSI searcher can be any unstructured, free-text file. In the context of this work, the document is a structured metadata document compliant with ISO 19115 standard. To avoid the waste of indexing time, content within the repeated occurring open and close tags "Title", "Abstract", "Science Keyword", "GCMD Keyword", "Location Keyword" and "Lineage" in the metadata documents were extracted to produce a compact document.

The semantic indexing process takes places when metadata documents are loaded into the catalog. Figure 3 displays this workflow. Steps 1–4 adopt the same steps a LuceneSearcher takes. At Step 5, a LSI searcher checks if a count matrix already exists. "No" indicates that this is the first time data has been inserted into the catalog, so a new count matrix will be created (Step 6.2). When importing more datasets, the count matrix needs to be updated (Step 6.1). For both cases, new keywords need to be extracted, new metadata documents

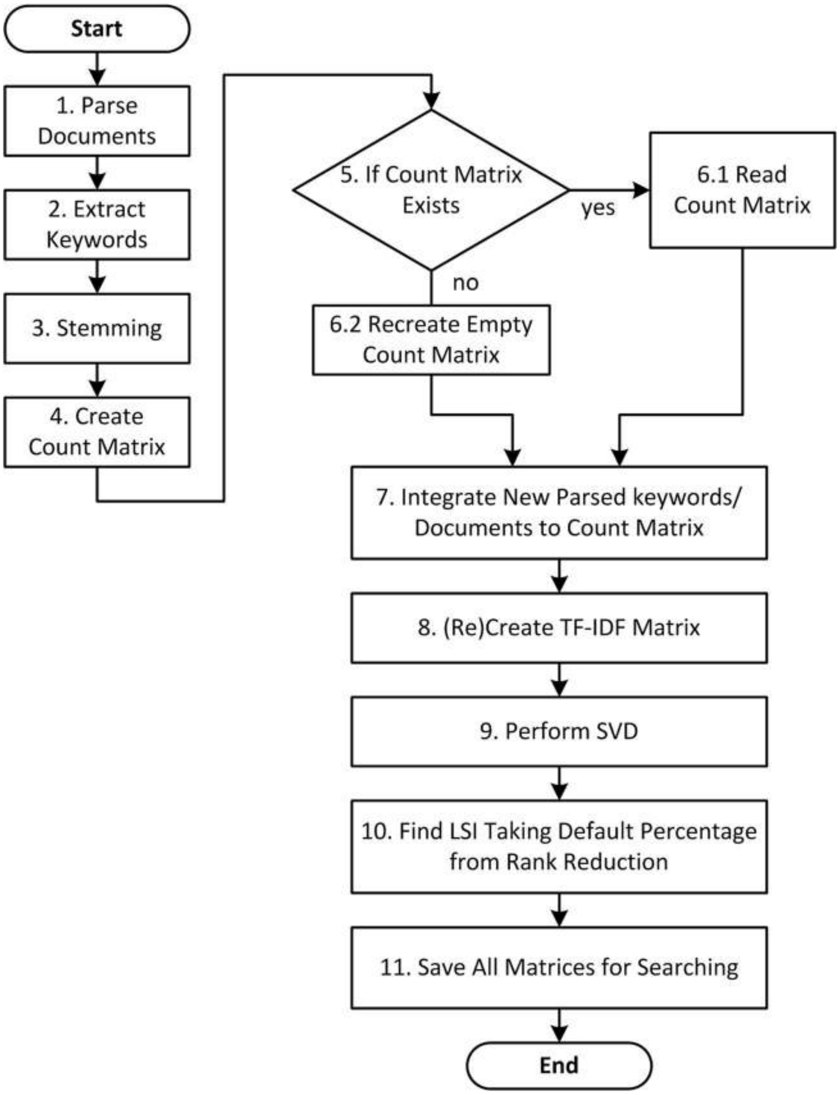

Start
1. Parse Documents
2. Extract Keywords
3. Stemming
4. Create Count Matrix
5. If Count Matrix Exists
yes
6.1 Read Count Matrix
no
6.2 Recreate Empty Count Matrix
7. Integrate New Parsed keywords/ Documents to Count Matrix
8. (Re)Create TF-IDF Matrix
9. Perform SVD
10. Find LSI Taking Default Percentage from Rank Reduction
11. Save All Matrices for Searching
End

Fig. 3 Workflow to create and update semantic matrix

need to be indexed, and the cell value of the count matrix needs to be computed. The count matrix, either empty or not, will be converted to a nested hashmap, or a hashmap of a hashmap, represented below:

$$\text{Map} < \text{keyword ID}, \ \text{Map} < \text{document ID}, \ \text{frequency} \ >>$$

The root hashmap uses unique keywords in the count matrix as keys, and child maps as values. In a child map, a document ID is the key and the frequency of a given keyword in the given document is the value. This data structure makes the update of count matrix very efficient. At Step 7, when there is new metadata inserted in the database, the keywords inside each metadata file will be extracted and the nested hashmap will be updated. If a key has already existed for a certain keyword, only a new entry<documentID, count>in the child map needs to be created. Otherwise, both root and child map need to create new keys to account for this metadata update. After this process, at Step 8, a new TF-IDF matrix is performed on the updated count matrix and then a SVD is performed at Step 9 on top of the TF-IDF matrix for LSI modification. At Step 10, the LSIsearcher uses the default rank preservation rate to create and save (Step 11) a new LSI matrix for the semantic search, discussed in previous section.

Experimental results

This section compares the search performance for the LSIsearcher and LuceneSearcer for polar data harvested from the Global Change Master Directory Portal for the Arctic Region (http://gcmd.nasa.gov/KeywordSearch/Home.do?Portal=arctic). The dataset covers a wide range of topics, from marine environment, geology to hazard. Table 3 lists the sample queries for performance evaluation. As known, recall and precision are two primary factors to evaluate search

Table 3 Selected Queries

Number	Queries
1	offshore hydrocarbon
2	offshore hydrocarbon mining
3	Marine biology data
4	Frozen ground Canada
5	Sediment data Yukon
6	Canada air pilot
7	Earthquake magnitude data Arctic
8	Sea level increase climate

performance. Recall measures the ability of a search tool in pulling relevant records from a database. Mathematically, it can be represented as:

$$recall = \frac{number\,(m)\,of\,relevant\,records\,returned\,by\,a\,search\,tool}{total\,number\,(n)\,of\,relevent\,records\,in\,a\,database} \tag{2}$$

As it is a very time consuming process to manually check all records and grab all relevant records (the denominator) to compute recall rate, in our experiment, instead of computing and comparing the actual recall rates of the two search methods, we introduced a modified measure, termed recall speedup, for the evaluation purpose. The mathematical form of recall speedup is:

$$recall_{speedip} = \begin{cases} \dfrac{number\,(m1)\,of\,relevant\,records\,returned\,by\,LSI\,searcher}{number\,(m2)\,of\,relevant\,records\,returned\,by\,Lucene\,Searcher}, (m2 > 0) \\ m1,\ (m2 = 0) \end{cases} \tag{3}$$

For any query posed to the same database, the denominator of recall, in Eq. (2), should be the same. The recall speedup is in fact equal to the ratio between the recall rate of LSIsearcher and the recall rate of the LuceneSearcher. A speedup value more than 1 means that LSIsearcher leads to a better recall than a LuceneSearcher. A value less than 1 means that LuceneSearcher performs better than LSIsearcher. If the value equals 1, both searchers achieve same performance in terms of recall.

Figure 4 demonstrates the recall speedup chart made by LSIsearcher for the selected queries. The dash-filled bars (rightmost bars) have value of all ones since the results from LuceneSearcher are normalized to 1. Some queries do not have these bars, i.e., Q2, Q3 and Q8, because there were no relevant results returned by LuceneSearcher on the given set of keywords. For those, the speedup recall of LSISearcher is to compare $m1$ with 1. Bars with solid colors in Figure 4 represent the speedup of LSISearcher for finding relevant records of a query to the LuceneSearcher. Different colors represent the degrees of rank reductions in the proposed semantic search. Though different rank reduction strategies have varying performance in terms of recall, they all perform much better than LuceneSearcher. As the rank preservation rate (the percentage value) decreases, the recall rate shows an increasing trend. Overall, the recall rate is highest when the rank preservation rate is at 70 %. Likewise, when this rate falls to 60 %, the recall search performance starts to decline. This is straightforward when the rate is small, more latent semantics are to be exploited. Therefore, more relevant records in the database can be identified. Using a small rank reduction rate vale means that less original semantics are preserved. Hence, the returned search results may include more irrelevant data than when using a higher rate. This behavior will to some extent lower the recall rate. For example, using LSISearcher with the query Q2 “offshore hydrocarbon mining”, users tend to find the distribution of offshore hydrocarbon resources and the mining activity. Whereas, using LuceneSearcher, no results were returned. At a rank preservation rate of 60 % for LSIsearcher, eight records are returned, but only four of them are relevant. Therefore, according to our strategy, the recall speedup is 4.0. When the rank preservation rate is set at 70 %, only six records are returned, but five of them are relevant. The recall speedup is therefore 5.0, higher than the recall speedup at 60 % rate.

Very different results were returned at different rank preservation rates too. When the rate is set to 60 %, irrelevant records such as “Canadian Active Control Point Observational Data” and “National Permafrost Database in Canada” were returned. This is due to that linkage among terms and concepts are established upon very loose semantic associations. While at 70 % rank reduction rate, relevant records such as “Frontier Well Data for North of 60—National Energy Board, which was not found at 60 % ratio, were returned as relevant results.

To measure the number of relevant records in the returned result, another performance factor—precision is used. Precision compares the number of relevant records in the result over all records returned by a search tool. In Eq. (4), precision is defined mathematically:

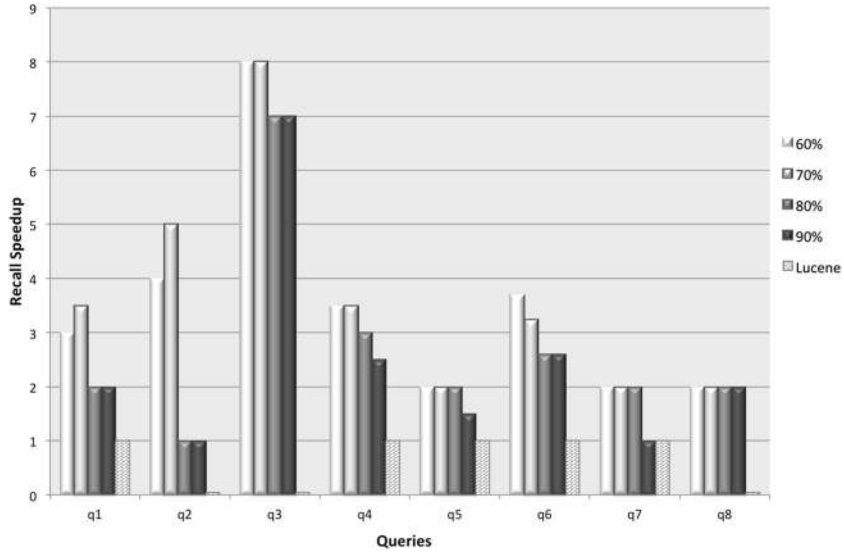


Fig. 4 Speedup of recall by LSISearcher at different rank preservation rates (solid grey bars) over LuceneSearcher (*dash-filled bars*)

$$precision = \frac{number\,(m)\,of\ relevent\,records\,in\,the\,result}{number\,of\ results\,by\,a\,search\,tool} \quad (4)$$

Figure 5 demonstrates the precision rates of LSISearcher and LuceneSearcher. We again investigated the precision achieved with varying rank preservation rates for LSIsearcher. In general, when a recall rate of a search tool increases, the precision will always decrease. This is because high recall rates rely on more relevant records to be found by expanding the search, either by searching keywords that have similar meanings to replace the original keywords or adding those that are semantically related. But when search criteria are loosened to increase recall, more irrelevant results will be returned, affecting the result precision. Through analyzing our experimental results, though, we found that LSIsearcher performs better or as well as LuceneSearcher on five of the eight given queries, including Q2, Q3 and Q3, which receive no result from the Lucene searcher. This good precision benefits greatly from the proposed ranking mechanism which introduces an improved version of similarity measure as well as an appropriate cut off value to filter out irrelevant data results. Another interesting finding is that, as the level of rank preservation rate becomes lower (or percentage value goes lower), the precision reduces. This is because when a semantic relationship is identified more through latent semantics than original semantics, more noise and undesirable artifacts may be introduced, therefore affecting the overall performance. The optimal value for rank preservation in our search context is 90 %. Below we provide detailed analysis on query results based on this value.

For Query 1, “offshore hydrocarbon”, the user tends to get data about the hydrocarbon data, such as oil, gas or petroleum. For LuceneSearcher, four records are returned. Since two of them include datasets that can be used to assess potential

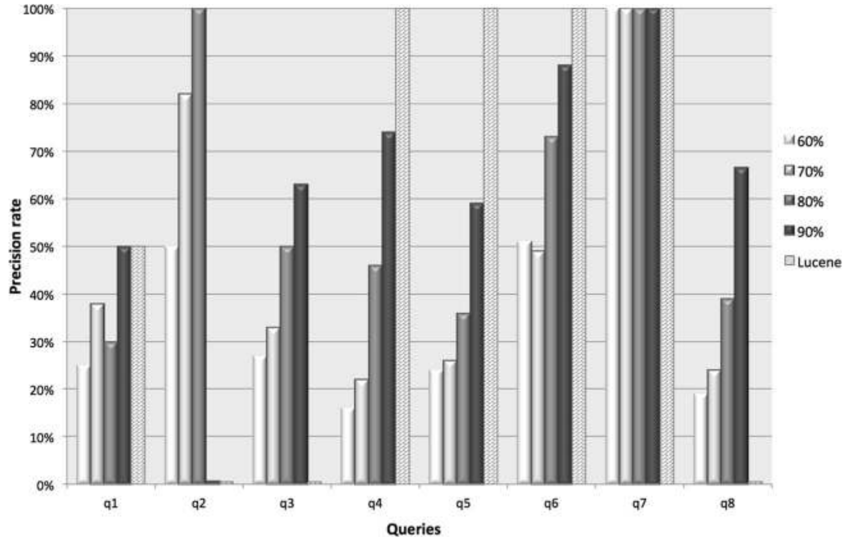


Fig. 5 Comparison of precision between LSISearcher and LuceneSearcher. (*solid grey bars*) over LuceneSearcher (*dash-filled bars*)

location of hydrocarbon resources but are not the actual locations for these resources, such as “Frontier Seismic Line Location Data”, they are ranked as irrelevant. Therefore, the LuceneSearcher receives 50 % precision rate. For the proposed semantic searcher, the number of returned results doubled that from the LuceneSearcher. As the results include data that is related but does not specify the inclusion of data about “offshore hydrocarbon”, such as “Internet GIS Geoscience data compilation”, they are considered as irrelevant as well. Based on this strict criterion, four of the eight results are classified as relevant, resulting in a precision of 50 %. LSISearcher ties LuceneSearcher on this query, but it finds more relevant records (4 vs. 2) than the LuceneSearcher found.

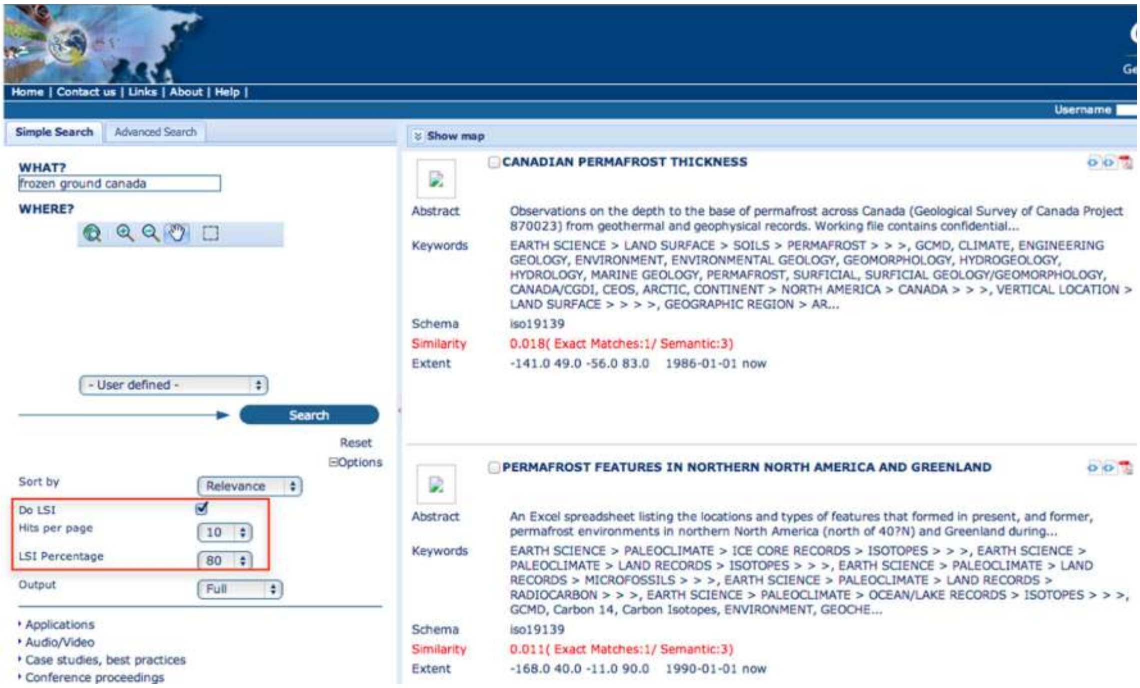
Using Q3 “Marine biology”, as another example, the full text Lucene searcher is not able to find any relevant data. However, our semantic searcher is able to find eleven related datasets, seven of which are actually relevant based on a manual verification. This result demonstrates the improved performance achieved by our semantic searcher.

For Query 8 “Sea level increase climate change”, the proposed semantic searcher returns three data records and two of them are directly related datasets. One record about “Marine Reservoir Effect” contains data about the impact of glaciation, climate and sea level change to Arctic molluscs, but not the data recording the sea level change over years due to changing climate. Therefore, although interesting, it is deemed irrelevant. In this case, the precision of our semantic searcher is 67 %.

The above analysis shows that, the semantic searcher performs much better in terms of recall rate than full-text search tool Lucene. Its precision is certainly better than Lucene for queries that cannot be answered by Lucene, and the precision of semantic searcher is almost as good as Lucene for other queries.

Graphic user interface

Figure 6 displays the search result for Query 4 “Frozen Ground Canada” in the GUI of the enhanced Geonetwork, which now supports both full text search and the proposed semantic search. When the “do LSI” box on the left hand panel is unchecked, the search uses the default Lucene searcher. When it is checked, the semantic search is performed at a specified rank preservation rate. In Fig. 6, this rate is set at 80 %. Users have the option to choose from a list of rank preservation rates to experiment with the data. The right hand result panel includes the title, abstract, science keywords, schema, geographic extent and similarity of matched records. The similarity, in red, shows the similarity score ranked by the proposed technique. Besides the numerical scores, the GUI also shows the number of exact matched keywords and the number of semantically matched keywords. For a full text-based search tool such as Lucene, a metadata document must contain all search keywords (by exact match) to be considered as relevant. Contrary to this, for a semantic search tool such as the proposed LSI searcher, a metadata document will be returned if it contains keywords either matching exactly or



WHAT? frozen ground canada

- User defined -

Sort by: Relevance

Do LSI ☑

Hits per page: 10

LSI Percentage: 80

Output: Full

□ CANADIAN PERMAFROST THICKNESS

Abstract: Observations on the depth to the base of permafrost across Canada (Geological Survey of Canada Project 870023) from geothermal and geophysical records. Working file contains confidential...

Keywords: EARTH SCIENCE > LAND SURFACE > SOILS > PERMAFROST > > >, GCMD, CLIMATE, ENGINEERING GEOLOGY, ENVIRONMENT, ENVIRONMENTAL GEOLOGY, GEOMORPHOLOGY, HYDROGEOLOGY, HYDROLOGY, MARINE GEOLOGY, PERMAFROST, SURFICIAL, SURFICIAL GEOLOGY/GEOMORPHOLOGY, CANADA/CGDI, CEOS, ARCTIC, CONTINENT > NORTH AMERICA > CANADA > > >, VERTICAL LOCATION > LAND SURFACE > > > >, GEOGRAPHIC REGION > AR...

Schema: iso19139

Similarity: 0.018(Exact Matches:1/ Semantic:3)

Extent: -141.0 49.0 -56.0 83.0 1986-01-01 now

□ PERMAFROST FEATURES IN NORTHERN NORTH AMERICA AND GREENLAND

Abstract: An Excel spreadsheet listing the locations and types of features that formed in present, and former, permafrost environments in northern North America (north of 40?N) and Greenland during...

Keywords: EARTH SCIENCE > PALEOCLIMATE > ICE CORE RECORDS > ISOTOPES > > >, EARTH SCIENCE > PALEOCLIMATE > LAND RECORDS > ISOTOPES > > >, EARTH SCIENCE > PALEOCLIMATE > LAND RECORDS > MICROFOSSILS > > >, EARTH SCIENCE > PALEOCLIMATE > LAND RECORDS > RADIOCARBON > > >, EARTH SCIENCE > PALEOCLIMATE > OCEAN/LAKE RECORDS > ISOTOPES > > >, GCMD, Carbon 14, Carbon Isotopes, ENVIRONMENT, GEOCHE...

Schema: iso19139

Similarity: 0.011(Exact Matches:1/ Semantic:3)

Extent: -168.0 40.0 -11.0 90.0 1990-01-01 now

Fig. 6 An enhanced GUI of Geonetwork. Text in red and text in red box are modifications for the semantic search component

semantically with the search keywords. But the number of matched keywords should be the same as the number of the search keywords considered relevant. For the record "Canadian Permafrost Thickness" in Fig. 6, it only contains one exact match to the search keywords, so it is considered irrelevant by Lucene. However, the keywords in the query can be matched semantically with this record, so the semantic searcher returns the document. This enhanced GUI not only shows the advantages of LSIsearcher, but also allows end users to experiment with the rank reduction ratio to determine the parameter that optimizes the search performance.

Conclusions and discussions

This paper reports our implementation of a semantic search technology based upon latent semantic analysis to improve search performance and make the dataset more discoverable in a metadata catalog. Polar science and polar data discovery is the primary application area of this proposed technique. The following intellectual contributions are made through this work: methodologically, we proposed a new ranking mechanism that combines a revised cosine similarity and a semantic filter to ensure the high precision of the search results. The semantic search technique makes it possible to learn latent semantics from data itself. The knowledge learned (containing the semantic associations among concepts and terms) has the advantages of being consistent with the actual semantics residing inside the data, leading to good search performance. We also experimented with different rank reduction mechanics and identified the optimized configuration to achieve high recall and precision for the semantic search tool. Practically, we incorporated the service-oriented design of an open source catalogue solution, Geonetwork, and successfully extended it to support the proposed semantic search function. This decision was made to reach the polar data community, since Geonetwork could be capitalized on as a widely used as catalogue solution for hosting geospatial data resources. Strategically, this search/data discovery component is an essential building block in the Polar CI development and it directly addresses the data discoverability challenge widely acknowledged in the polar research community (Pundsack et al. 2013).

In the future, we will design more experiments with polar data at a larger sample size and conduct interviews to obtain the actual recall rate of search results for a comprehensive evaluation of LSIsearcher's performance. By doing this, we can further evaluate the impact of expressiveness of latent semantics and the effectiveness of bottom-up semantic analysis approach to the search performance. This issue is a crux in intelligent search and is ubiquitously existed in both the top-down and bottom-up semantic search approaches. We will also extend the indexing framework to make it more scalable by introducing the distributed computing framework, such as cloud computing and Hadoop (Gao et al. 2014), to handle the search of big polar data. A third plan is to make the new search service OGC (Open Geospatial Consortium)-compliant and easily accessible through open-source strategy.

Acknowledgments This paper is supported by National Science Foundation Award #1349259 and Open Geospatial Consortium Award #027216.

References

Aguilar-Lopez D, Lopez-Arevalo I, Sosa V (2009) Usage of domain ontologies for web search, *International Symposium on Distributed Computing and Artificial Intelligence 2008* (DCAI 2008). Springer, pp. 319–328

Alhabashneh O, Iqbal R, Shah N, Amin S, James A (2011) Towards the development of an integrated framework for enhancing enterprise search using latent semantic indexing, *Conceptual Structures for Discovering Knowledge*. Springer, New York, pp 346–352

Beran B (2007) Hydroseek: an ontology-aided data discovery system for hydrologic sciences. Citeseer

Bernard L, Einspanier U, Haubrock S, Hubner S, Kuhn W, Lessing R, Lutz M, Visser U (2003) Ontologies for intelligent search and semantic translation in spatial data infrastructures. *Photogrammetrie Fernerkundung Geoinformation*, 451–462

Bhogal J, Macfarlane A, Smith P (2007) A review of ontology based query expansion. Inf Proc Management 43:866–886

Budak Arpinar I, Sheth A, Ramakrishnan C, Lynn Usery E, Azami M, Kwan MP (2006) Geospatial ontology development and semantic analytics. Trans GIS 10:551–575

Castells P, Fernandez M, Vallet D (2007) An adaptation of the vector-space model for ontology-based information retrieval. Knowl Data Eng IEEE Trans 19:261–272

Celikyilmaz A, Hakkani-Tur D, Tur G (2010) LDA based similarity modeling for question answering, *Proceedings of the NAACL HLT 2010 Workshop on Semantic Search. Association for Computational Linguistics*, pp. 1–9

Chen H, Martin B, Daimon CM, Maudsley S (2013) Effective use of latent semantic indexing and computational linguistics in biological and biomedical applications. Front Physiol 4:8

Christidis K, Mentzas G, Apostolou D (2012) Using latent topics to enhance search and recommendation in enterprise social software. Expert Syst Appl 39:9297–9307

Cimiano P, Haase P, Heizmann J (2007) Porting natural language interfaces between domains – a case study with the ORAKEL system –. In: *Proceedings of the International Conference on Intelligent User Interfaces* (IUI), pp. 180–189

Cochran PA (2013) Impacts on indigenous peoples from ecosystem changes in the Arctic Ocean, *environmental security in the Arctic Ocean*. Springer, New York, pp 75–79

Daniel C, Wood FS (1999) Fitting equations to data: computer analysis of multifactor data. John Wiley & Sons, New York

Deerwester S et al (1990) Indexing by latent semantic analysis. J Am Soc Inf Sci 41(6):391–407

Dhillon IS, Fan J, Guan Y (2001) Efficient clustering of very large document collections, *data mining for scientific and engineering applications*. Springer, New York, pp 357–381

Dumais ST (2004) Latent semantic analysis. Annu Rev Inf Sci Technol 38:189–230

Fernández M, Cantador I, López V, Vallet D, Castells P, Motta E (2011) Semantically enhanced information retrieval: an ontology-based approach. Web Semant Sci Serv Agents World Wide Web 9:434–452

Gao S, Li L, Li W, Janowicz K, Zhang Y (2014) Constructing gazetteers from volunteered big geo-data based on Hadoop. Comput Environ Urban Syst. doi:10.1016/j.compenvurbsys.2014.02.004

Goelzer H, Huybrechts P, Loutre M-F, Goosse H, Fichefet T, Mouchet A (2011) Impact of Greenland and Antarctic ice sheet interactions on climate sensitivity. Clim Dyn 37:1005–1018

Gosling S, Taylor R, Arnell N, Todd M (2011) A comparative analysis of projected impacts of climate change on river runoff from global and catchment-scale hydrological models. Hydrol Earth Syst Sci 15: 279–294

Harvey F, Kuhn W, Pundt H, Bishr Y, Riedemann C (1999) Semantic interoperability: a central issue for sharing geographic information. Ann Reg Sci 33(2):213–232

Hjørland B (2010) The foundation of the concept of relevance. J Am Soc Inf Sci Technol 61:217–237

Holland MM, Bitz CM, Tremblay B (2006) Future abrupt reductions in the summer Arctic sea ice. *Geophysical Research Letters* 33

Hyvönen E, Saarela S, Viljanen K (2004) Application of ontology techniques to view-based semantic search and browsing. In the semantic web: research and applications. Springer, Berlin Heidelberg, pp 92–106

Janowicz K (2012) Observation-driven geo-ontology engineering. Trans GIS 16:351–374

Jones CB, Abdelmoty AI, Finch D, Fu G, Vaid S (2004) The spirit spatial search engine: architecture, ontologies and spatial indexing, *geographic information science*. Springer, New York, pp 125–139

Li W, Yang C, Raskin R (2008a) A semantic enhanced search for spatial web portals. AAAI Spring Symp Tech Rep SS-08–05:47–50

Li W, Yang P, Zhou B (2008b) Internet-based spatial information retrieval. In Encyclopedia of GIS, pp. 596–599, Springer US

Li W, Yang C, Nebert D, Raskin R, Houser P, Wu H, Li Z (2011a) Semantic-based web service discovery and chaining for building an Arctic spatial data infrastructure. Comput Geosci 37:1752–1762

Li Z, Yang CP, Wu H, Li W, Miao L (2011b) An optimized framework for seamlessly integrating OGC web services to support geospatial sciences. Int J Geogr Inf Sci 25:595–613

Li W, Goodchild MF, Raskin R (2012) Towards geospatial semantic search: exploiting latent semantic relations in geospatial data. Int J Digit Earth. doi:10.1080/17538947.2012.674561

Li W, Li L, Goodchild MF, Anselin L (2013) A geospatial cyberinfrastructure for urban economic analysis and spatial decision-making. ISPRS Int J Geo-Inf 2:413–431

Liu K, Yang C, Li W, Li Z, Wu H, Rezgui A, Xia J (2011) The GEOSS clearinghouse high performance search engine. In Geoinformatics, 2011 19th International Conference on (pp. 1–4). IEEE

Lopez V, Pasin M, Motta E (2005) Aqualog: An ontology-portable question answering system for the semantic web. In: Gómez-Pérez A, Euzenat J (eds) ESWC 2005. LNCS, vol. 3532. Springer, Heidelberg, pp 546–562

MacKenzie CM, Laskey K, McCabe F, Brown PF, Metz R, Hamilton BA (2006) Reference model for service oriented architecture 1.0. OASIS Standard 12

Mangold C (2007) A survey and classification of semantic search approaches. Int J Metadata Semant Ontologies 2(1):23–34

Marshall J, Armour K, Scott J, Ferreira D, Shepherd TG, Bitz CM (2013) The ocean's role in polar climate change: asymmetric Arctic and Antarctic responses to greenhouse gas and ozone forcing

Nicholls RJ, Marinova N, Lowe JA, Brown S, Vellinga P, De Gusmao D, Hinkel J, Tol RS (2011) Sea-level rise and its possible impacts given a 'beyond 4 C world'in the twenty-first century. Philos Trans R Soc A Math Phys Eng Sci 369:161–181

Overpeck J, Hughen K, Hardy D, Bradley R, Case R, Douglas M, Finney B, Gajewski K, Jacoby G, Jennings A (1997) Arctic environmental change of the last four centuries. Science 278:1251–1256

Pundsack J, Bell R, Broderson D, Fox GC, Dozier J, Helly J, Li W, Morin P, Parsons M, Roberts A, Tweedie C, and Yang C (2013) Report on workshop on cyberinfrastructure for polar sciences. St. Paul, Minnesota. *University of Minnesota Polar Geospatial Center*, 17pp

Ramachandran R, Movva S, Graves S, Tanner S (2006) Ontology-based semantic search tool for atmospheric science, *Proceedings of 22nd International Conference on Interactive Information Processing Systems for Meteorology, Oceanography, and Hydrology*, http://ams.confex.com/ams/Annual2006

Rose L (2004) Geospatial portal reference architecture: a community guide to implementing standards-based geospatial portals. *OpenGIS Disscusion Paper*, OGC, 04–039

Scudellari M (2013) An unrecognizable Arctic, Global climate change. NASA, Greenbelt, MD. http://climate.nasa.gov/news/958

Singhal A (2001) Modern information retrieval: a brief overview. IEEE Data Eng Bull 24:35–43

Skedsmo M, Taylor F, Palmer O, Guomundsson M (2011) Arctic Spatial Data Infrastructure (SDI): Pan-Arctic Cooperation among Ten Mapping Agencies. Available from: http://132.246.11.198/2012-ipy/Abstracts_On_the_Web/pdf/IPY2012ARAbstract01950.pdf

Spearman C (1904) The proof and measurement of association between two things. Am J Psychol 15:72–101

Steinbach M, Karypis G, Kumar V (2000) A comparison of document clustering techniques, *KDD workshop on text mining*. Boston, pp. 525–526

Stouffer RJ, Yin J, Gregory J, Dixon K, Spelman M, Hurlin W, Weaver A, Eby M, Flato G, Hasumi H (2006) Investigating the causes of the response of the thermohaline circulation to past and future climate changes. *Journal of Climate* 19

Tran T, Cimiano P, Rudolph S, Studer R (2007) Ontology-based interpretation of keywords for semantic search. Springer, Berlin Heidelberg, pp 523–536

Wang H (2013) Distributed catalogue search of earth observation data. George Mason University

Xiong J, Huang W, Jin C (2009) An ontology-based semantic search approach for geosciences, *Knowledge Acquisition and Modeling, 2009. KAM'09. Second International Symposium on. IEEE*, pp. 87–90

Zimov SA, Schuur EA, Chapin FS III (2006) Permafrost and the global carbon budget. Sci (Wash) 312:1612–1613