Call for Short Papers: OGRS 2016
http://2016.ogrs-community.org/

Implementing an open source spatio-temporal search platform for Spatial Data Infrastructures

Paolo Corti, Harvard Center for Geographic Analysis
Benjamin Lewis, Harvard Center for Geographic Analysis
Tom Kralidis, Open Source Geospatial Foundation
Ntabathia Jude Mwenda, Harvard Center for Geographic Analysis

Abstract

A Spatial Data Infrastructure (SDI) is a framework of geospatial data, metadata, users and tools intended to provide an efficient and flexible way to use spatial information. One of the key software components of an SDI is the catalogue service which is needed to discover, query and manage the metadata. Catalogue services in an SDI are typically based on the Open Geospatial Consortium (OGC) Catalogue Service for the Web (CSW) standard which defines common interfaces for accessing the metadata information.

A search engine is a software system capable of supporting fast and reliable search, with features such as full text search, natural language processing, weighted results, fuzzy tolerance results, faceting, hit highlighting and many others. In this paper we will be focusing on the Lucene, a powerful Java-based search library.

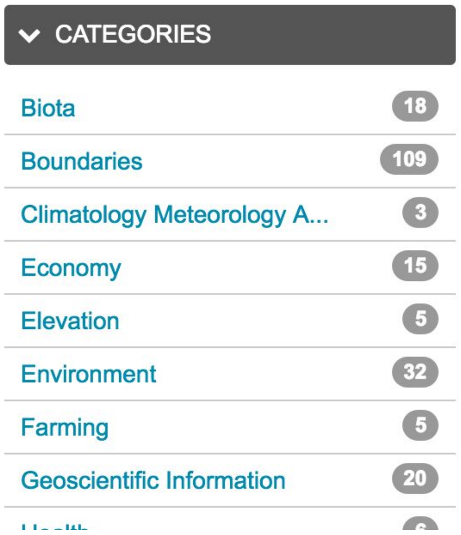
The Centre of Geographic Analysis (CGA) at Harvard University is trying to integrate within its public domain SDI (WorldMap http://worldmap.harvard.edu), the benefits of both worlds (OGC catalogues and search engines).

Harvard Hypermap (HHypermap) is a component that will be part of WorldMap, built on an open source stack. The system implements an OGC catalogue based on pycsw, to provide access to metadata in a standard way, and uses a search engine based on Solr/Lucene, to provide advanced search features typically found in search engines.

The need for a search engine in a SDI

Typically the way a search engine works can be split in two distinct phases: indexing and searching. During the indexing phase all of the documents (metadata, in the SDI context) that must be searched are scanned, and a list of search terms (an index) is built. For each search term, the index keeps track of the identifiers of the documents that contain the search term. During the searching phase only the index is being looked at, and a list of the documents containing the given search term is quickly returned to the client.

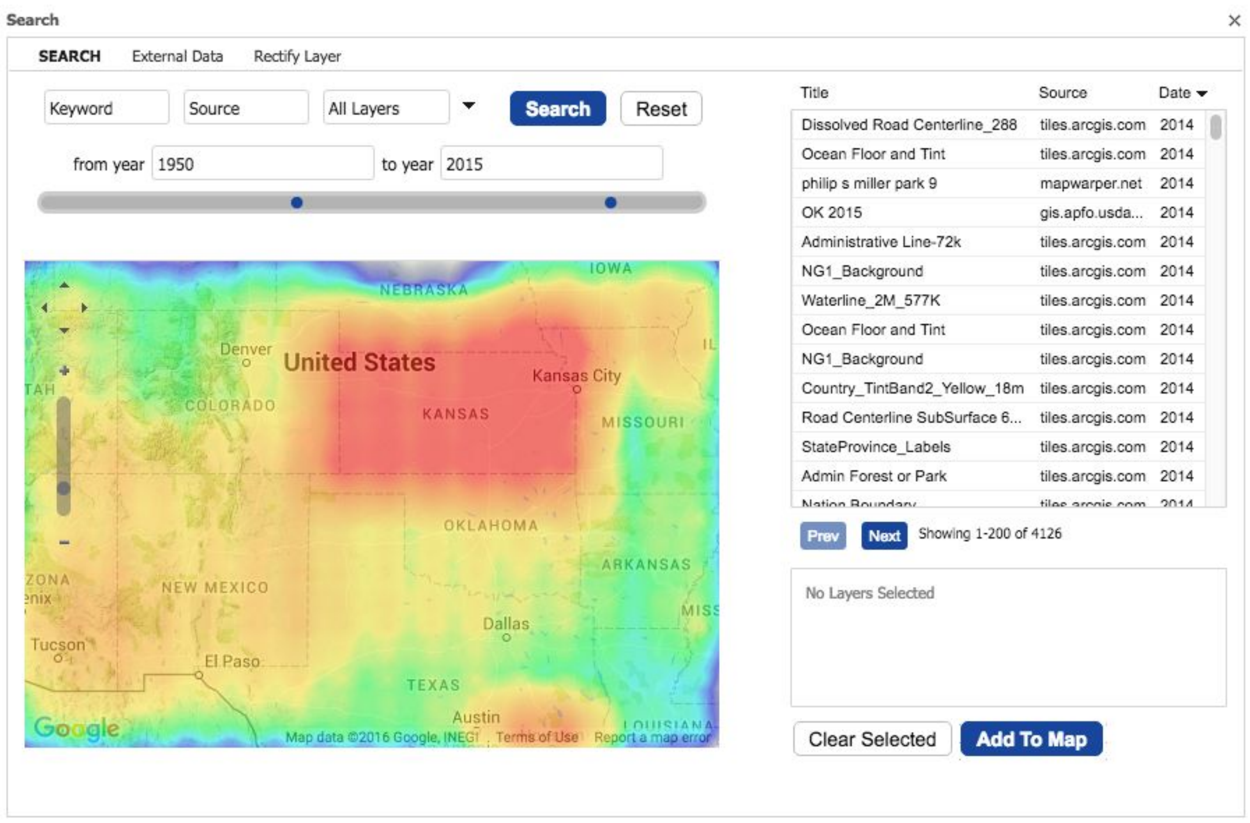
This way of working make a search engine extremely fast in outputting results. On top of this, a search engine provides many other useful features, improving dramatically the experience of users searching for materials.
Most notably, search engines are able to handle the ambiguities of natural languages, thanks to *stop words* (words filtered out during the processing of text), stemming (ability to detect words derived from a common root), synonyms detection, and controlled vocabularies such as thesauri and taxonomies.
It is possible to do phrase searches and proximity searches (search for a phrase containing two different words separated by a specified number of words). Results can be weighted, providing a way to provide users more relevant results. In addition it is possible to use regular expressions, wildcard search, and fuzzy search to provide results for a given term and its common variations.
It is also possible to support boolean queries: a user is able to search results using terms and boolean operators such as AND, OR, NOT.
Hit highlighting is an interesting feature that provides immediate suggestions to the user typing the text to search in metadata.
Another search engine feature that can be extremely useful for searching a metadata catalogue is faceted search. Faceting is the arrangement of search results in categories based on indexed terms. This capability makes it possible for example to provide an immediate indication of the number of times that common keywords are contained in different metadata documents.
A typical use case is with metadata categories, keywords and regions. Thanks to facets the user interface of the SDI catalogue can display counts for documents by category, keyword or region.



CATEGORIES

Biota	18
Boundaries	109
Climatology Meteorology A...	3
Economy	15
Elevation	5
Environment	32
Farming	5
Geoscientific Information	20

Some search engines can support temporal and spatial faceting, two features that are extremely useful for browsing geospatial metadata.
Dates faceting will display the number of documents by date range. Space faceting will provide the number of layers or features for a given spatial extent. In the following figure a heatmap is generated by spatial faceting which shows the distribution of layers for a given geographic region.



Title	Source	Date
Dissolved Road Centerline_288	tiles.arcgis.com	2014
Ocean Floor and Tint	tiles.arcgis.com	2014
philip s miller park 9	mapwarper.net	2014
OK 2015	gis.apfo.usda...	2014
Administrative Line-72k	tiles.arcgis.com	2014
NG1_Background	tiles.arcgis.com	2014
Waterline_2M_577K	tiles.arcgis.com	2014
Ocean Floor and Tint	tiles.arcgis.com	2014
NG1_Background	tiles.arcgis.com	2014
Country_TintBand2_Yellow_18m	tiles.arcgis.com	2014
Road Centerline SubSurface 6...	tiles.arcgis.com	2014
StateProvince_Labels	tiles.arcgis.com	2014
Admin Forest or Park	tiles.arcgis.com	2014

Showing 1-200 of 4126

No Layers Selected

Furthermore, from a software engineering perspective, search engines are highly scalable and replicable, thanks to their shardable architecture.

Integration of a CSW catalogue with a Search Engine

There can be two approaches to integrating a search engine and the CSW catalogue within an SDI:

- combining the best of both worlds
- using the search engine as a backend for the CSW catalogue

HHypermap in this first phase combines the best of the two worlds.
Clients willing to search the catalogue provided by HHypermap in a standard way will be able to run a search using the OGC CSW endpoint as well as other standards-based search APIs (OpenSearch, OAI-PMH, SRU)
Clients willing to use the more advanced features, such as faceting or full text features such as synonyms detection, fuzzy tolerance, weighted search, etc. will be able to run a search accessing the restful API provided by the search engine.
Harvard CGA is currently looking into the possibility of enabling the search engine itself to be a backend to its CSW endpoint. This way it will be possible to access at least a part of the powerful search engine features from within the CSW itself.

Harvard Hypermap: an SDI search engine based on a open source stack

HHypermap is an application that manages OGC web services (such as WMS, WMTS), Esri REST endpoints and other types of map service harvesting, maintaining uptime statistics for services and layers. The aim of HHypermap is to provide a more effective search experience to WorldMap users and for users outside WorldMap. WorldMap is an open source mapping platform developed by the CGA to lower barriers for scholars who wish to explore, visualize, edit and publish geospatial information.
HHypermap is exclusively built on a open source software architecture. The web application runs on the Django web framework which is written in Python. Relational structured information is stored in PostgreSQL. Remote services and layers are harvested with Celery, an asynchronous Python task queue, taking advantage of the RabbitMQ message broker. pycsw is a Python library which provides the OGC CSW interface and other standards-based APIs. HHypermap can use Solr or ElasticSearch as a search engine, both of which are based on Lucene.

HHypermap source code can be found here: https://github.com/cga-harvard/HHypermap