Additional text in "Notes" view

Alexandria Digital Library Project

Four Steps to Geospatial Enlightenment

Greg Janée

Tammy said she wanted to hear "the five steps to geospatial enlightenment." Well, here are four. You can decide if they constitute enlightenment or not.

Topics

- A few things we've learned
- Issues that affect all three roles
  - producer
  - broker/accumulator
  - consumer
- The ADL solutions

For the most part ADL has focused on the broker/accumulator role, but there are issues described herein that apply to all roles.

ADL solutions will be mentioned only in passing; the focus of this presentation is on issues that any project dealing with geospatial/georeferenced data is likely to encounter.

Geospatial discovery

- Can't beat word search when it works
  - I want a map of Boulder
  - ∃ "Downtown street map of Boulder, Colorado"
- But there are so many names for a place...
  - Boulder, Arapahoe County, Colorado
  - Chautauqua, Mapleton Hill, Pearl Street Mall
  - Area code 303, ZIP code 80305, UTM grid 13S
  - Flatirons, Rocky Mountains, Front Range
  - Landers earthquake, hurricane Hugo

The naïve approach to geospatial discovery is to do direct word search over whatever (textual) placenames are present in the metadata.

This is unreliable because so many names apply to any given place: names at many different scales; political, physiographic, and technical names; formal and informal names; event names; etc.

If you're still not convinced...

- Remote-sensing imagery is nameless
  - "AVHRR NOAA-13 2002-06-03 14:33 UTC"
- Challenge: exactly which two words will find a USGS map of the Flatirons behind Boulder, Colorado?

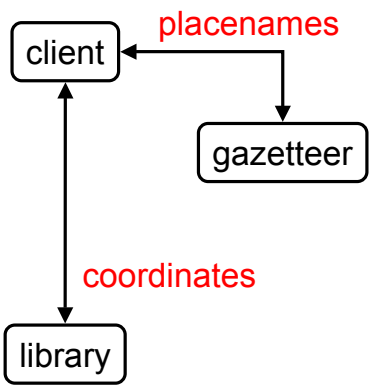
Eldorado Springs

Remote-sensing imagery (including aerial photography) is naturally nameless, even title-less (the title displayed above is manufactured from other metadata). Word-based placename search is useless here.

Names are also often entirely unexpected. Answer to challenge: the USGS uses a system of named quadrants to index their maps, and most if not all of their data. So one and only one name describes an entire area. How can you know what name that is?

ADL approach

- Coordinate-based representation and discovery
  - generic lat/lon coordinates
  - rich geometry
    - **polygons, polylines**
  - spatial operators
    - **overlaps, contains**
- Gazetteer
  - *content standard* defines representation
  - *service* maps placenames ↔ coordinates



For reliable discovery, some kind of coordinate-based system must be used instead of text search, or at least integrated into a text-based discovery system. ADL's approach is not the only possibility, but it does nicely fit the metadata we've encountered and the functionality users want.

Notice that the gazetteer is *not* tightly integrated with the library itself, as it conceivably could be. See next.

Gazetteers: necessary evil

- Few (public) sources of gazetteer data
- Lousy quality
  - digitized from maps
- Difficult problems
  - conflation
  - classification
  - boundary determination
  - change over time
- Conclusion
  - gazetteer-based spatial reasoning seems unlikely
  - interaction will likely remain client-centric

Data sources: Who has gazetteer data? What's their motivation for publishing it? The ADL gazetteer is derived from just two government sources.

Quality: placenames are often digitized off of maps, so the location of a place is where the label appeared.

Conflation: when integrating multiple gazetteer sources, determining if two places are the same is surprisingly difficult, and requires various kinds of heuristic/fuzzy matching.

Conclusion: if gazetteer data were complete and perfect, some sophisticated spatial reasoning could be performed on it, and the processing could live inside the library. Instead, it seems more likely that gazetteers and their fuzzy, imperfect data will stay close to clients, i.e., something for human eyeballs to evaluate.

Implications of data types

- Text is effectively typeless
  - text ≡ byte string
- Adding geospatial type (i.e., data types) has many implications:
  - input validation
  - internal structures, external representations
  - query language and processing
  - ranking
  - user interface components

Coordinate-based search means adding a notion of data types to the system.

Text search is amazing when you consider how incredibly effective it is despite its simplicity. When we say "text" we really mean "byte string": typically, no validation or structuring is done on text (no semantic analysis, no grammar or spell checking, not even checking for non-printable characters). Search over text amounts to simple pattern matching. Text search over multiple fields, and boolean combinations of text constraints, can be implemented using simple text search, i.e., no special processing is required.

Adding (complex) data types changes the situation completely. Now validation is required, as are complex representations and structures. It must be possible to express and perform typed query constraints and boolean combinations of different types of constraints. Type-specific ranking methods are required, as are methods for combining different types of rankings. Specialized user interface components are required to input and to view typed data.

ADL approach

- Discovery: buckets
  - extensible data type system for metadata
    - **XML representations**
    - **search operations**
  - explicit metadata mappings
  - foundation for collection-level statistics
  - 9 Dublin Core-like standard buckets
- Baby steps:
  - spatial ranking
  - user interface components
    - **map-based result, item viewers**
    - **manual georeferencing tool**

Discovery: ADL supports data types in the form of "buckets." A *bucket* is a typed index over a collection's item-level metadata; its definition includes standard representations and search operations. ADL has defined 6 bucket types (geospatial, temporal, etc.), but the scheme is extensible. ADL does not mandate any particular metadata scheme, but instead supports explicit mappings of metadata to buckets. Search is performed over buckets, over individual metadata fields mapped to buckets, and over collection statistics derived from buckets. ADL has defined 9 standard buckets, analogous to Dublin Core, to support cross-collection discovery.

The single most important feature that distinguishes ADL's approach from many others is the incorporation of data types.

Baby steps: ADL has played with spatial ranking methods, and has developed some alpha-level UI components.

Scalability

- Easy to accumulate lots of data
  - satellites image continuously
- Text
  - inverted indexes scale amazingly well
- Geospatial
  - R-trees scale... not so well
    - **indexing becomes unwieldy at 10^6 items**
  - combining spatial, other constraint types is difficult
    - **efficiently, that is**

We have found it easy to accumulate lots of geospatial data, both in terms of total size and in terms of number of items.

Text search scales amazingly well; after all, an inverted index's size (in terms of number of distinct words) is essentially constant after some point.

Geospatial search should scale well in theory, but we have encountered many problems indexing as few as a million items. Indexing seems to require many gigabytes of RAM and days of processing.

How to *efficiently* combine radically different types of constraints (geospatial and textual, for example), either within a relational database engine or across multiple external search services, seems to be an open research problem. We have encountered many problems in this area.

ADL approach

- Distributed library
- Federated item-level search
  - over buckets
  - over individual metadata fields mapped to buckets
- Centralized collection-level search/ranking
  - over collection statistics
    - **derived from bucket mappings**

ADL's fundamental approach to handling large quantities is to distribute the problem.

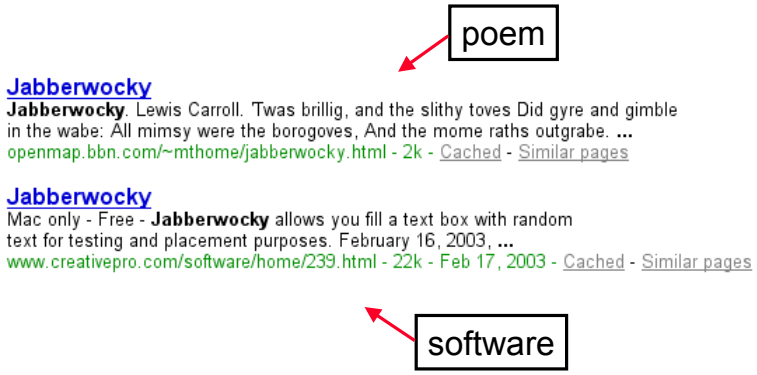
Item-level search is fundamentally federated.

Collection-level search (really more of a ranking of collections) is centralized. The search is performed not over item-level metadata, but over collection-level statistics (e.g., spatial coverage histograms) derived from item-level metadata.

Thus geospatial discovery in ADL is a two-step process: identify relevant collections, then search those collections. The collection-level and item-level query languages are coordinated so that this can be collapsed into a one-step process if desired.

Textual context

- Effective context in text is easy to provide
- Consider:

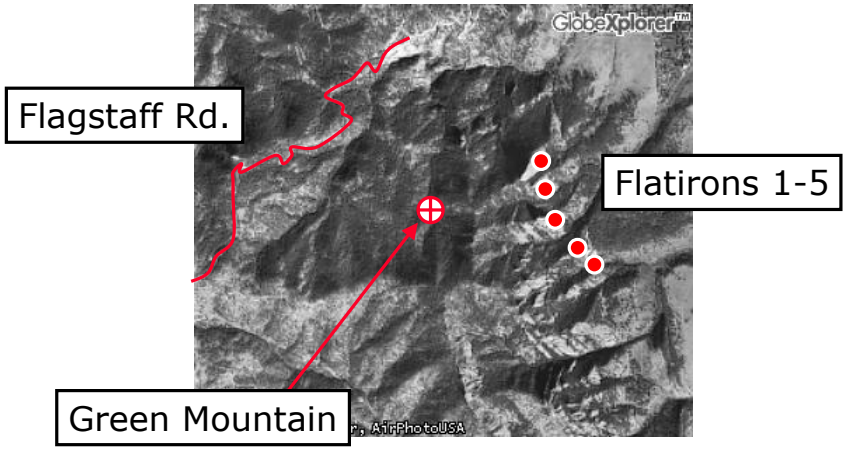


Users need context of where they are in an information space and what the library is about in order to formulate queries, interpret results, and evaluate individual library items.

Textual context is simple and effective: consider the above example. Using only a handful of surrounding words and a few square inches of screen space, it is possible to completely understand the very different nature of these two returns to the query "Jabberwocky."

Geospatial context

- Does this answer your question?



Geospatial context is even more critical. Context is necessary to be able to formulate queries. (Consider how useless interfaces are that ask users to manually enter coordinates.)

Context is also necessary not only to use geospatial items, but just to evaluate them. Consider the above aerial photograph as a response to a query about a place, say Green Mountain. Without the annotations, it is useless.

ADL doesn't have a solution to this issue yet. Our plan is to integrate some lightweight GIS functionality (something like Berkeley's GISviewer) into the library.

Summary

- Searching by placenames is unreliable
  - must use coordinate system
  - gazetteers are a necessary evil
- Coordinates introduce the need for data types
  - ease of text vanishes
  - need validation, data type-specific tools
- Scalability is a concern
  - easy to accumulate lots of data
  - difficult to combine spatial with other constraint types
- Library must provide geospatial context
  - to form queries
  - to evaluate/use item