

CSCI 572 HOMEWORK 3 REPORT

CHANDRASHEKAR CHIMBILI

chimbili@usc.edu

JAYAVANTH SHENOY

jshenoy@usc.edu

MADAN PATIL

madanpat@usc.edu

SREEPADA RAO SINGEETHAM

singeeth@usc.edu

How easy to use was d3?

It's a very robust library for manipulating svg. The built-in functions are pretty useful. The collection of d3 visualizations helped us to start off with the library. The collections covered all visualizations that we were looking for.

- Sunburst: We chose this visualization because it inherently represents hierarchy. In our visualization, the innermost layer identifies locations, next layer identifies year and the outermost layer identifies most prominent keywords in documents that year.
- Dendrogram: This chart gives a very clear representation of the link graphs that we have implemented.
- Bubblechart: It visualises the prominence of a discrete fields in a global context. Each bubble represents a science keyword and the size of the keyword indicates its prominence.
- Treemap: The representation of data as rectangles offer better readability and size estimation compared to other graphs.
- Indented Tree: This provides a more traditional representation of json data.
- Bar graph: A standard choice for graphical representations of counts of a chosen facet.

How easy to use was Banana?

Banana uses Kibana's powerful dashboard configuration capabilities, ports key panels to work with Solr, and provides significant additional capabilities, including new panels that leverage D3.js. Banana was created to visualise data stored in Solr, which makes any content retrieved as Solr index eligible for banana visualization. This makes panel addition to banana straightforward as they accept the Solr indexed fields directly as input.

The histogram panel can now be represented by indicating the row field (year) and the column field (location) as direct inputs from Solr Index fields, signifying the ease of use of banana.

How easy to use was FacetView?

As the facetview github repository points out, it does not support SOLR anymore. The developers point out that the need to maintain SOLR support was less and hence the functionality was totally removed. They suggested to add a new function named *solrsearchquery()* instead.

However after going through the history of FacetView we found that the old version, still supports querying and displaying from SOLR. We found the code in bitbucket. We had to

change the CSS style sheets so that User Interface is coherent across d3, banana and Facetview.

We just had to modify "search_url" in jquery.facetview.js to point it to local SOLR. Also default filters were changed to location, keywords and year. The default query was *:*

Note that the search in Facetview is real time. The query fired gets data by setting default facets. The data displayed is aggregated by location, year and keywords facets. Also we can filter selecting a particular facet. Result now filters over given query and selected facet. This is quite interesting and very helpful. This supports slice and dice features which are supported by many analytics solutions.

The FacetView also displays the resulting documents which satisfy the query. There is flexibility to choose what attributes and features of a document to display by default. Due to time constraint we did not modify any of those configurations. Overall FacetView is a very good tool and easy to configure and use.

What was the hardest part, loading data, or visualizing it?

D3 was not built with Solr indexes in context nor was Solr built with JSON results as priority. This rises the problem of representation of Solr index data as JSON parsable by D3. Visualization by itself using D3 on static data is not an easy task. Considering we have to dynamically parse the Solr Index to D3 compatible JSON, loading the data was the hard part. Each chart plays around with svg and has different JSON structure which make it practically impossible to write a single parser to satisfy all the D3 visualizations we can come up with. In each visualization we had to modify the query to get an intermediate JSON which was then parsed into the json D3 graph requires.

The D3 community provides lot of examples on how to incorporate the charts and the instructions to use them were well documented. This solves one of the challenges with D3, i.e., visualization of data. This makes the major challenge of implementing D3 for our Solr result representation only the parsing and loading of data.

Conclusion

Banana, D3 and FacetView are a great ways to visualize Solr results with minimal effort. The easy to understand functionality and less effort on visualization of data is an advantage of these tools.

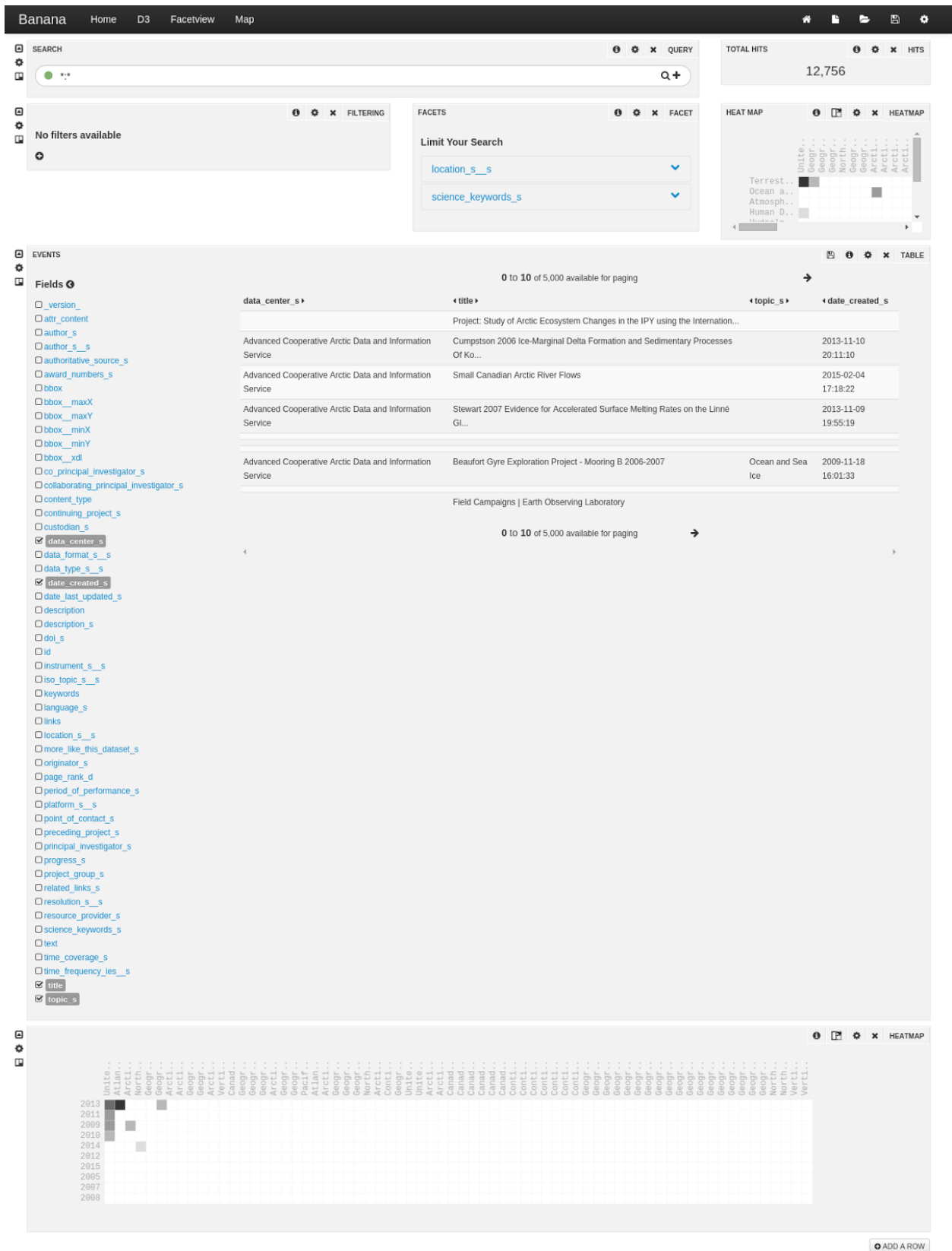
This is the link to the video demo of the visualizations and the demo:

<https://youtu.be/rld7uVCETC8>

Appendix: These are some of the screenshots of our visualizations. Questions 3a - 3d are answered as part of the video.

Banana

Screenshot:



Facet

View

Screenshot

Facetview

Home D3 Banana Map

SEARCH

Find

FILTERS

location_s_s

science_keywords_s

results 0 to 10 of 163

Cumpston 2006 Ice-Marginal Delta Formation and Sedimentary Processes Of Kongsvegen Glacier in Kongsfjorden, Svalbard, Norway

Sea Ice Age Depict Fundamental Changes in Ice Cover | EOSDIS - Earthdata Website

Ice Camp Position/Velocity(FINAL) (ASCII) [Moritz, R.]

Design and Initialization of an Ice-Tethered Profiler Array Contributing to the Arctic Observing System

Bacterial and extracellular polysaccharide content of brine-wetted snow over Arctic winter first-year ice

Project: Collaborative Research: A pan-Arctic, storm-by-storm isotopic investigation of the influence of Arctic sea ice on precipitation - a crucial link in the coupled climate system

How do I use the new data services in the Reverb search and order tool? : National Snow & Ice Data Center

Lisa

Arctic Sea Ice: binary

Reverb Tutorial : National Snow & Ice Data Center

Molly

Arctic sea ice on the wane: Now what? | EOSDIS - Earthdata Website

Map Screenshot

