CSCI 572: Assignment 3

Visual Search and Interaction with NSF and NASA Polar Datasets

Harshit Gulati

Divydeep Agarwal

Jugpreet Singh Talwar

Nikhil Moorjani

Ruhi Agarwal

In this assignment, we leveraged the Apache Solr index and the D3.js, Banana and FacetView data visualization technology (http://d3js.org/) to interact and visualize search engine data.

- **How easy to use was D3?**

  D3 is a JavaScript library that uses digital data to drive the creation and control of dynamic and interactive graphical forms which run in browser.

  D3 was comparatively easier as:

1. Compared to other software's, D3 is easily available to download and the README.md provides precise instructions.
2. It is a widely successful visualization platform and there are a lot of tutorials available as examples to give a good head start and hence can be learned with the use of web.
3. It is easy to understand as it is web-based.
4. The visualizations are simple to understand. Hence can be used for a larger audience base, increasing reachability.
5. It is flexible and offers a wide variety of data formats for data visualization.
6. It works well with existing web technologies which makes it portable and easy to render.
7. The visualizations have an aesthetic feel and are very attractive and rich in display.

- **How easy to use was Banana?**

  Visualization using Banana: It's a time series approach for data stored in Solr. We designed a new dashboard to add various panels based on our dataset from a list of available panels. We specified the fields from our dataset which were required to be visualized on our dashboard. We visualized both time-based and non-time based visualization. Various panels which we implemented and were suitable for our dataset include Hits, Terms, Heatmap, Scatterplot, Tagcloud, etc.

Tradeoffs while using Banana:

1. Banana was easy to download and install.
2. It can directly talk to Solr. We added it as a web-app for the Tomcat web server & configured it to use the required collection for visualization.
3. 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.
4. We needed to configure the new dashboard to be used as the default one for our browser on selecting the banana tab.
5. The configuration involved selecting the correct collection for visualization and specifying the rows containing various panels.
6. We can even filter the query results based on certain facets and it's a very convenient method to display results.

Overall, the Banana visualization is easy to use and it is helpful in visualizing the data as per user defined fields including both time-based and non-time based fields.

- **How easy was it to use FacetView?**

FacetView is a pure JavaScript frontend for ElasticSearch search indices.

Thoughts on ease of use:

1. The documentation and the code samples were given for ElasticSearch and there was no precise documentation to use FacetView with Solr.
2. It was not easy to adapt the code since changing the code required knowledge of not only Solr fields but also the corresponding mapping in ElasticSearch which was time consuming.
3. The javascript file had functions written according to ElasticSearch and hence the return values as well as the parameters to be passed had to be converted to fit Solr’s query model.
4. The JSON output of SOLR is very complex and hence decoding the right structure and plugging in FacetView was a challenging task.
5. FacetView also does not have any in-built support for filtering the queries and hence we had to write our own method of linking the documents to the facets.
6. The FacetView offers a very elegant user-interface and is simple to understand and best for queries where filtering will be used.
7. The learning curve for this was the most.

- **Creating the map:**

  1. We used the D3 visualization Datamaps (datamaps.io) to create a world map layer and then plot our documents based on latitude/longitude combination.
  2. We created a custom JSON to be fed into the datamaps function. Once the map gets loaded in the SVG container the points were plotted on the map.
  3. The json data was present in SOLR. We created a Java servlet which manipulated the JSON data for our custom format.
  4. We then set the various parameters within Datamaps function to get a better control over the visualization.
  5. For the challenge queries, we used the query and function query fields of SOLR.
  6. The data was plotted using latitude, longitude, document identifier and the point color and radius.

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

  The hardest part was loading data as different visualizations required different data formats. There was extra processing of data to clean it and convert into JSON/CSV format which was required by D3, FacetView. The structure was well-defined but building it was a complex task. Visualizing data was comparatively simple once we had the data in the right formats as input.

- **{Extra credit 1}**

  Making FacetView work with Solr. (ExtraCredit1.docx)
  We have documented the process for making FacetView work with Solr. FacetView by default supports ElasticSearch. The document can serve as a good reference point for integrating both the tools.

- **{Extra credit 2}**

  Using multi-valued attributes (metadata) to determine facets in faceted search using FacetView. (ExtraCredit2.docx)
  FacetView currently supports only single-valued attributes in metadata field. Our metadata in SOLR was multivalued because the way it was tagged by us in homework 2. This made us deriving a novel approach for making facetview work with multivalued attributes. We have made changes to the javascript function of facetview.

- **Demo Link**: http://youtu.be/A5aI8r_Ib3E