Bahador Saket and Jackson Toeniskoetter
CS 544
Prof. Carlos Scheidegger
Final Project Report
May 13, 2015

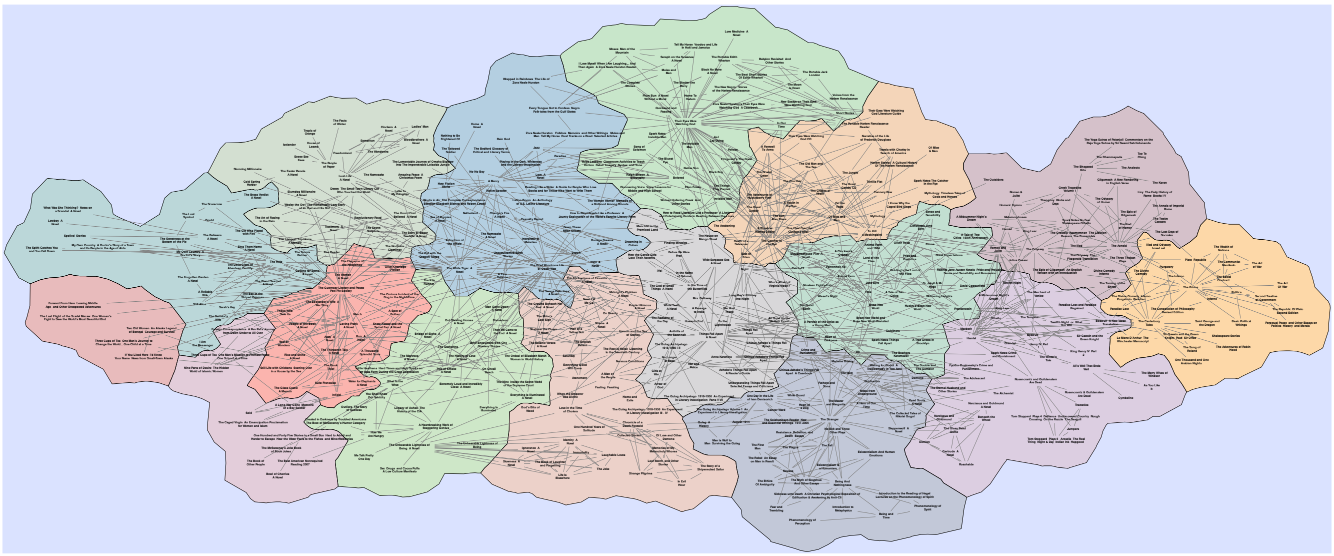

Figure 1. Representation of 300 nodes and about 700 links with GMap visualization; underlying data from the Book dataset [4].

What is GMap?

GMap is a an algorithm for producing graph visualizations which take the form of geographic maps, see Figure 1. Many graph visualizations inadequately display the higher level structure (e.g. clustering, neighborhoods) implicit in large datasets. GMap attempts to overcome these flaws explicitly representing such structure by geographic proximity and boundaries between countries. Such maps can be more aesthetically appealing than traditional node-link diagrams.

First published in 2010 by Gansner et al. A year later, the algorithm was applied to produce map-based visualizations of large-scale dynamic relational data. In 2013, the website http://gmap.cs.arizona.edu/ was created to make an implementation freely available for public use [4].

GMap first performs a clustering and a layout step to compute an initial drawing of the data set. Geographic boundaries are then inserted to separate the clusters based on connectivity in the original graph. A modified Voronoi diagram of the nodes is obtained from the layout and clustering steps, which is then used to generate the boundaries [1].

By visiting http://gmap.cs.arizona.edu/, you can see informally how GMap helps to visualize large datasets using a geographic metaphor [4].

Advantages of GMap for visualization

Previous user experiments indicate that participants perform tasks more accurately and quickly using map-based visualizations than node-link visualization [2]. Additionally, there is evidence that map-based visualizations are more appealing and engaging to participants, and that the data itself is more memorable when presented as a map [3].

Flaws of GMap

Despite the advantages of GMap when compared to node-link diagrams, it still has many flaws which need to be addressed. For instance, on data sets of 200 nodes, it took users 24 seconds to perform simple tasks such as path finding in sparse networks, and for denser networks the accuracy dropped and the required time increased substantially (10 seconds) [2]. This will clearly not scale to larger datasets with thousands or tens of thousands of nodes and links. As you can see in Figure, large diagrams become a dense blur of overlapping links.

Although GMap can provide an informative overview, there is currently no way to interactively explore the map of the data. If the user of GMap wishes to explore alternative views of the data, they need to rerun GMap.

Data

We have available to us every data set found at http://gmap.cs.arizona.edu/datasets. This includes the co-authorship network among International Symposiums on Graph Drawing authors from 1994-2015, a network of 3204 books constructed using Amazon's "customer's who bought this also bought" information, and many more.

Each dataset is processed at http://gmap.cs.arizona.edu into an image file, which contains labeled elements corresponding to the nodes and links in the data. For our project, we choose to process each dataset as an SVG file, so that it can be conveniently manipulated using JavaScript and d3. Since the data is presented in a clean format, we currently perform no actual cleaning of the data.

In Table 1, you can see an example of the Dot file obtained from the GMap project as the data source. In our visualization, we do not make use of the Dot file itself, but instead we use the SVG file generated by GMap.

```
graph {
  "0" [cluster="3", label="Drawing", pos="26.163,130.97"];
  "1" [cluster="3", label="Visualization", pos="270.82,243.14"];
  "2" [cluster="2", label="Graph", pos="271.43,16.263"];
  "3" [cluster="2", label="Arizona", pos="670.15,16.263"];
  "4" [cluster="2", label="University", pos="415.11,16.263"];
  "5" [cluster="1", label="Map", pos="513.44,131.26"];
  "0" -- "1";
  "1" -- "2";
  "0" -- "2";
  "3" -- "4";
  "5" -- "1";
  "5" -- "2";
}
```

Our current visualization makes use of the Book data set, but in principle any of the data sets will work.

Goals

While the static GMap visualization provides a nice overview of the data, we felt that it was necessary to be able to explore the data visually at a finer level of detail. To that end, we have implemented a web application which enables interactive features, allowing users to easily explore properties of user defined subsets of the data.

Features

Our system provides several methods to move around the visualization, to select subsets of the data, and to examine certain graph theoretic properties of these subsets, based on features such as graph distance and centrality.

Zoom/Pan: Our application's first improvement is to provide native zooming and panning over the map-based visualization, which accommodates the exploration of large and dense graphs.

Selections: Our system provides two ways to select subsets of the data. The first is by individually selecting nodes, either by mouse click or by querying the name. The second is by using a d3 brush to select every node in a region.

Shortest Path: Our application will highlight the shortest paths between every pair of currently selected nodes by pressing a button. This feature can be quite useful in evaluating how closely the map metaphor matches the data set. For example, in Figure 2, we show how two nodes which were placed very close to each other in the map have a relatively large graph distance. Furthermore, the shortest path moves away from both nodes and crosses itself in the embedding.

Our system provides two ways for users to select subsets of the data, either by brushing a region, or by individually selecting a group of nodes. After making a selection, users can identify multiple graph theoretic statistics about the selection, or they can identify connectivity properties, such as highlighting the shortest paths between every pair of nodes in the selection.

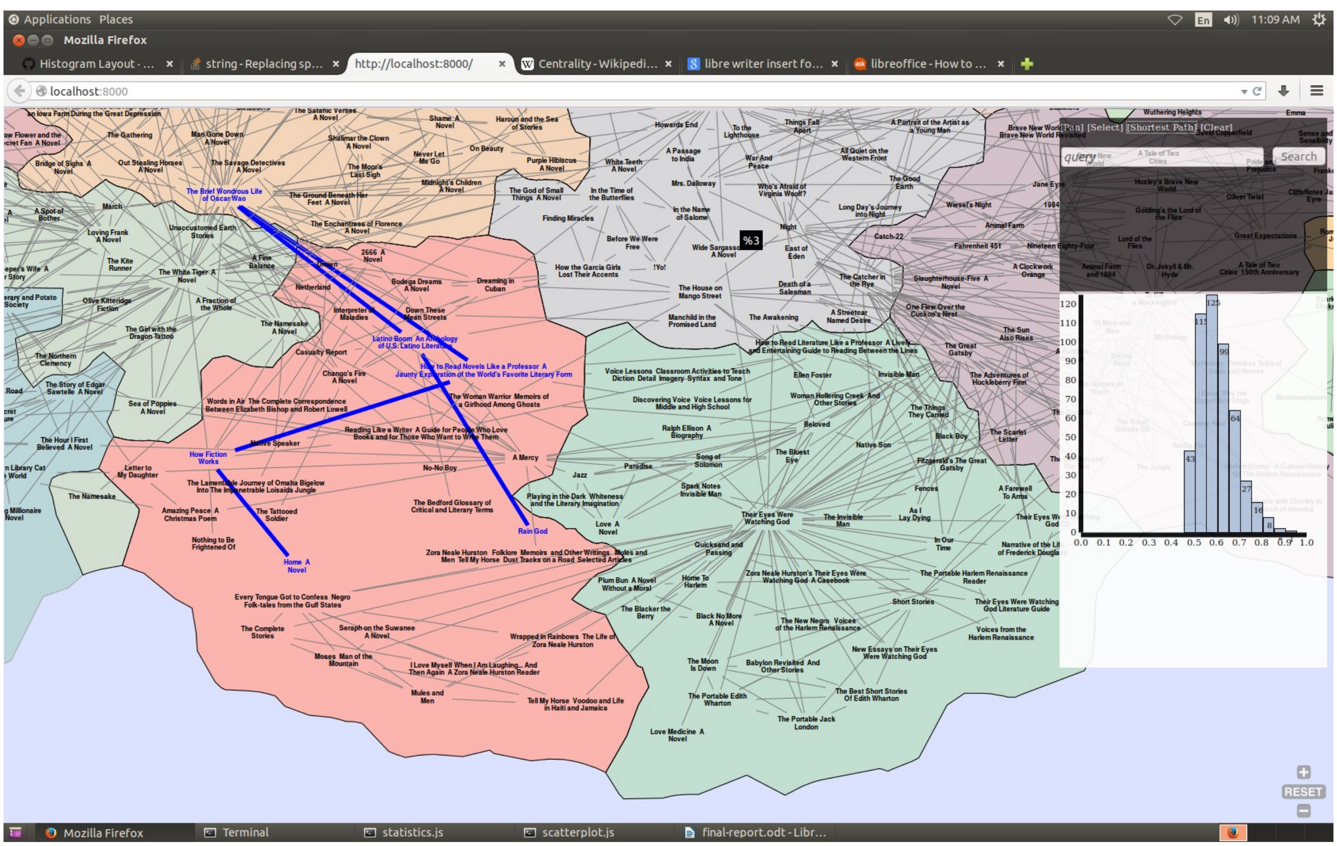

Figure 2. By highlighting the shortest path between these two geographically close nodes, we are able to identify a way in which the map metaphor fails.

Centrality: The visualization displays a histogram of the closeness centrality measure of each node in the graph. For a vertex v, centrality is defined as

$$Centrality(v) = \sum_{u} \frac{1}{d(v,u)}$$

for each vertex, we compute it's centrality, and divide by the maximum centrality so that the values fall in the range [0,1]. When the map is brushed, a second histogram is laid over the first, show the distribution of centrality values in the brushed region. The user can also brush the histogram, which highlights each node with a centrality value in the given range. See Figures 3 and 4 for examples.

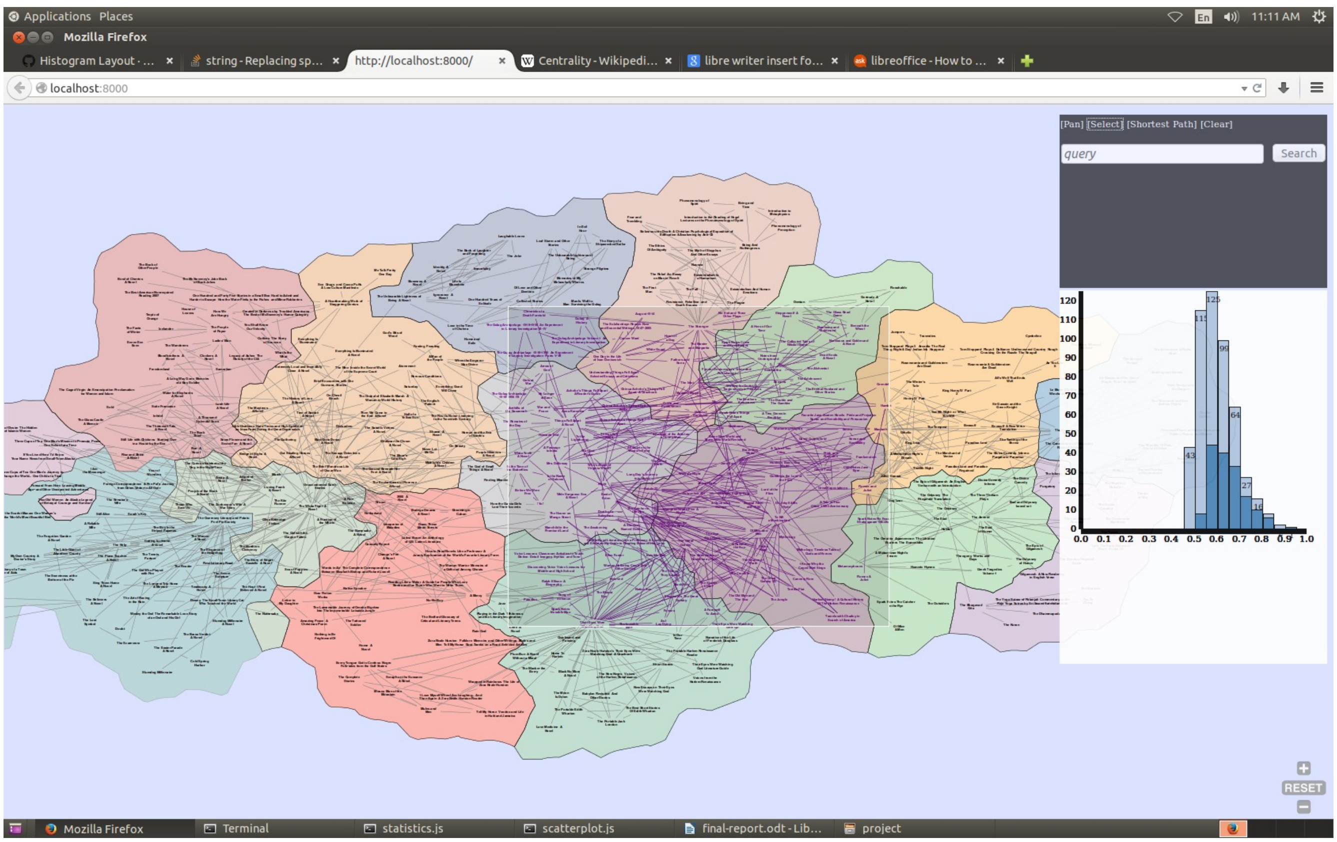

Figure 3: An example of how the centrality distribution of values in the brushed region is overlaid on the histogram of the centrality distribution for the entire graph.

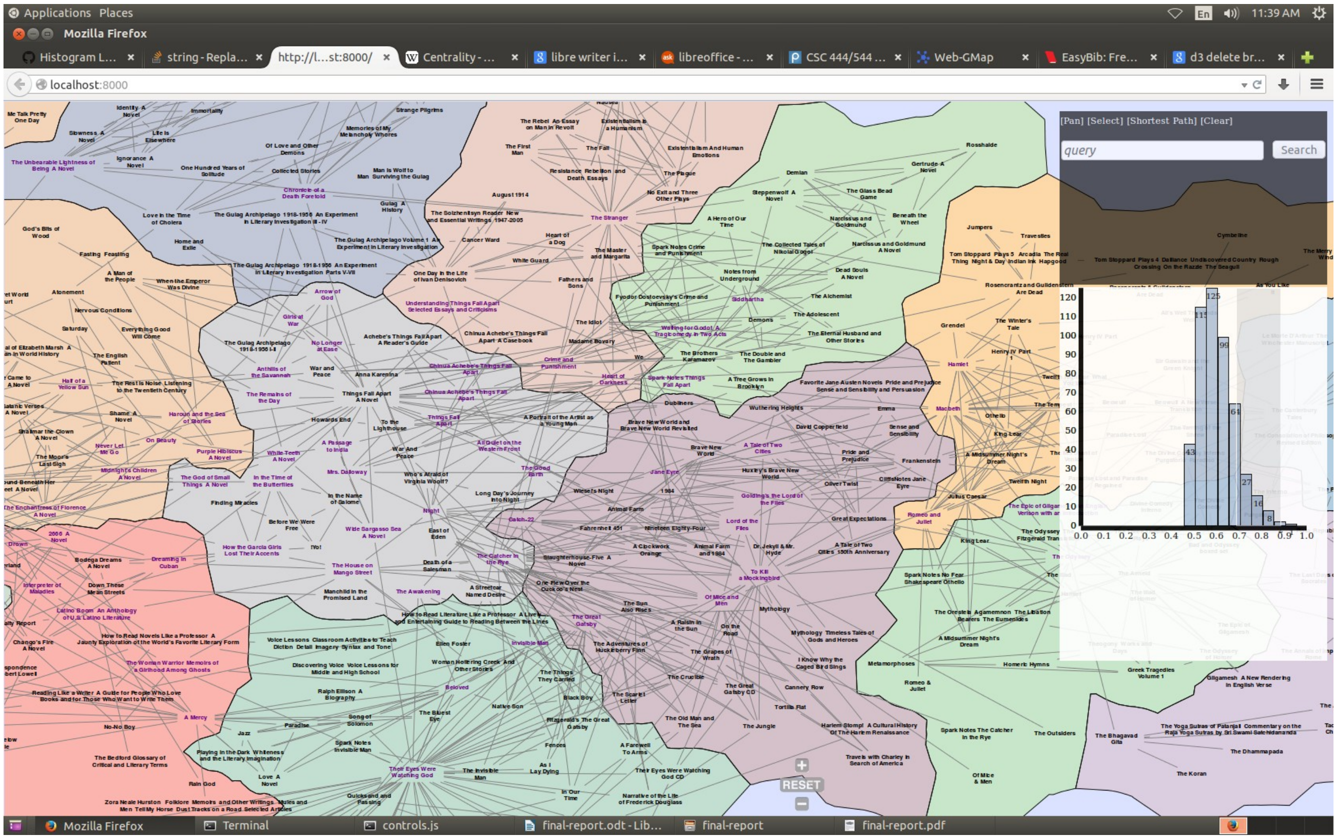

Figure 4: Example of brushing the histogram, and highlighting nodes whose centrality falls in the brushed range.

Limitations: We thought it would be easy to manipulate the SVG files provided by GMap and add interactivity. Unfortunately, there were several technical problems related to zooming in and out of the SVG, which caused the brushes to not rescale themselves when zooming was performed.

The visualization is somewhat laggy, probably resulting from a mismatch between d3's data structures and the graph library cytoscape.js which we used to implement graph theoretical algorithms.

Future Work

In the future, we would like to support multiple graph theoretic measures, and allow the user to select which ones to be displayed. Ideally, multiple measures could be compared at the same time for a single region in the visualization, such as the mutliple measures of centrality (see http://en.wikipedia.org/wiki/Centrality for examples) which differ from the closeness centrality we used.

Another direction is to support multiple brushes on the map, so that users could compare different regions of the graph. Unfortunately, d3 does not support this feature natively, so an implementation will have to find some way to freeze all but one brush, and to activate each brush on user demand.

References

[1] Gansner, Emden R., Yifan Hu, and Stephen Kobourov. "GMap: Visualizing graphs and clusters as

maps." *Pacific Visualization Symposium (PacificVis), 2010 IEEE*. IEEE, 2010.

[2] Saket B., Simonetto P., Kobourov S. G., Borner K., Node, Node-Link, and Node-Link-Group Diagrams: An Evaluation, In Proc. IEEE Symposium on Information Visualization **(**InfoVis**)**, 2014.

[3] Saket B., Scheidegger C., Kobourov S. G., Borner K., Map-based Visualizations Increase Recall Accuracy of Data, 17th International Conference on Eurographics Working Group on Data Visualization **(**EuroVis**)**, 2015.

[4] http://gmap.cs.arizona.edu/