Milestone Report

Authors: Kevin Wall, Mike Liu, Will Usher

Confirmation of Graph Structures

TODO: Kevin

Initial D3 Visualization

TODO: Mike

Parsing the DBLP Database & Scraping

One of the main challenges of our project was acquiring the data we wanted. Although the DBLP database provides a good consistent set of information about publications we were also hoping to include information about author affiliations which isn't in the database. This ends up being a pretty challenging web-scraping problem, in that this information is scattered across tons of different publisher web pages all with their own formatting for displaying the author affiliation data.

Fortunately the ACM publishes a few different journals that we can use all of which host information on the same site (dl.acm.org) which uses a consistent formatting for displaying this information, make it an easy candidate for scraping. We where also hoping to include IEEE TVCG as on of our journals but this journal actually has publications scattered on two different web pages and on one only shows affiliation information for the first author as far as we could tell. As a result of these challenges we'll likely just stick with the ACM published journals since we already have code together for scraping them.

Another challenge is simply dealing with the volume of data in the DBLP database. The entire database is in a single, massive XML file (1.6 GB!) and thus it must also be treated with some care. Since we only want a subset of journals for our final visualization we chose to generate JSON data files containing just the specific journals we're interested in and another file containing the list of authors in all the journals we've chosen. To generate this we use a fast streaming XML parser allowing us to process the dataset and build the JSON files for our selected journals and authors quickly and with low memory overhead. This parser is located in `dblp_to_json.py` and we use `scrape_affiliation.py` to handle scraping author affiliation information.

The output of our processor is a JSON file for each journal selected and a JSON file containing all the authors for all the selected journals. For example when running on the *Transactions on Graphics* journal our resulting journal data will be in `tog.json` and the author data stored in `authors.json`.