A Look into Successful Federal Legislation Bills

Yash Patel, Freddy Vallenilla, Allie Ai
yashpatel@fas.harvard.edu, fvallenilladorta@g.harvard.edu,
allie_ai@gse.harvard.edu

Final Project, CS171 (Spring 2014)

Group Members: Yash Patel (yashpatel@fas.harvard.edu), Freddy Vallenilla (fvallenilladorta@g.harvard.edu), and Allie Ai (allie_ai@gse.harvard.edu)

FINAL PROJECT PROCESS BOOK – CS171

TABLE OF CONTENTS

[This page intentionally left blank]

Initial Project Proposal

1. Background and Motivation. Discuss your motivations and reasons for choosing this project, especially any background or research interests that may have influenced your decision.

The American society is facing many challenges, social, political, and economic. To find solutions to current issues, the general public first needs to be informed of the political directions and decisions that the government is making. What decisions does the Congress make and how they make such decisions?

2. Project Objectives. Provide the primary questions you are trying to answer with your visualization. What would you like to learn and accomplish? List the benefits.

We are trying to answer the following questions: What decisions does the Congress make? Are there any clusters of issues that the Congress are primarily concerned with? Is there a party line (Democrats vs. Republicans) on those popular issues? If so, what are the patterns?

We would like to accomplish the following goals through our visualization product (i.e. the benefits of our product to our audience):

- Presenting information of political decisions made by the Congress to the general public in an accurate, visually appealing and easy-to-understand manner
- Explore and identity political dynamics existing in the Congress to gain deeper understanding of the functioning of the Congress

3. Data. From where and how are you collecting your data? If appropriate, provide a link to your data sources.

We will collect relevant data from Sunlight Data Services via API calls (http://sunlightfoundation.com/api/), in particular the Congress API http://sunlightlabs.github.io/congress/

4. Data Processing. Do you expect to do substantial data cleanup? What quantities do you plan to derive from your data? How will data processing be implemented?

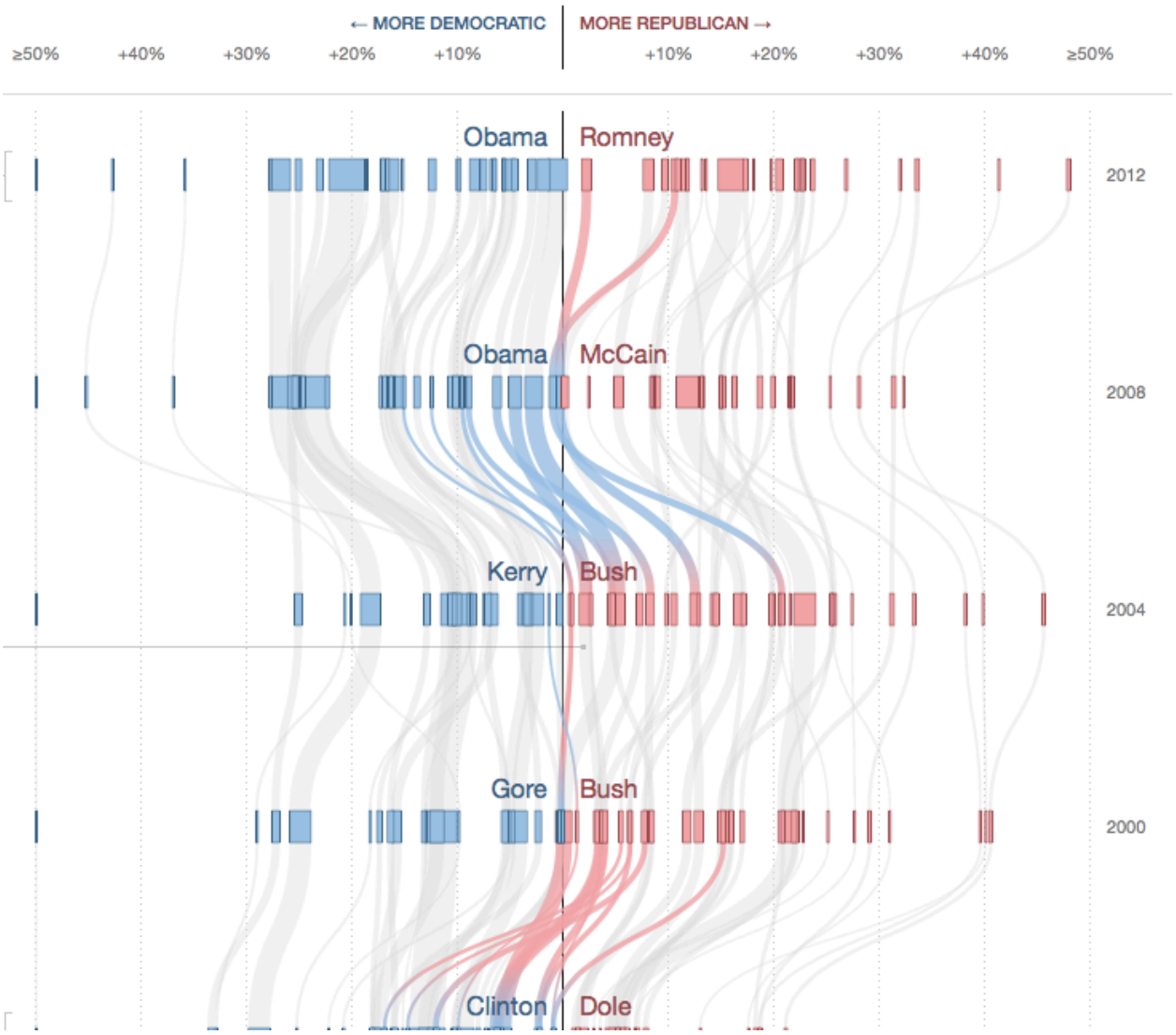
We do not expect to do substantial data cleanup because the API breaks information down into sub-categories that we can select from appropriately. We plan to use all the available data regarding Bills, Legislature, and the votes of each house. The data will be stored in arrays of objects via API calls. This way, this data remains up-to-date and will be used in the creation of the visualization.

5. Visualization. How will you display your data? Provide some general ideas that you have for the visualization design. Include sketches of your design.

We plan to display our data based upon the selected bill. An individual will have the ability to “mouseover” a bill and see which house/legislature voted in favor

of/against. We are hoping to integrate all the bills into a single, seamless visualization so that the users can simply scroll over all the bills and see the pertinent information.

A sample of what our visualization could look like upon completion is:



6. Must-Have Features. These are features without which you would consider your project to be a failure.

One of the most important features that we want for our visualization is a simple visual of how each Bill was voted for. This will allow the user to browse with ease and intuitively know how to go about using our visual. Also, our project should have up-to-date information. Providing information of previous years is useless and cannot be used in today's discussions.

7. Optional Features. Those features which you consider would be nice to have, but not critical.

- An implementation of Square's Crossfilter library to show the cross-section of

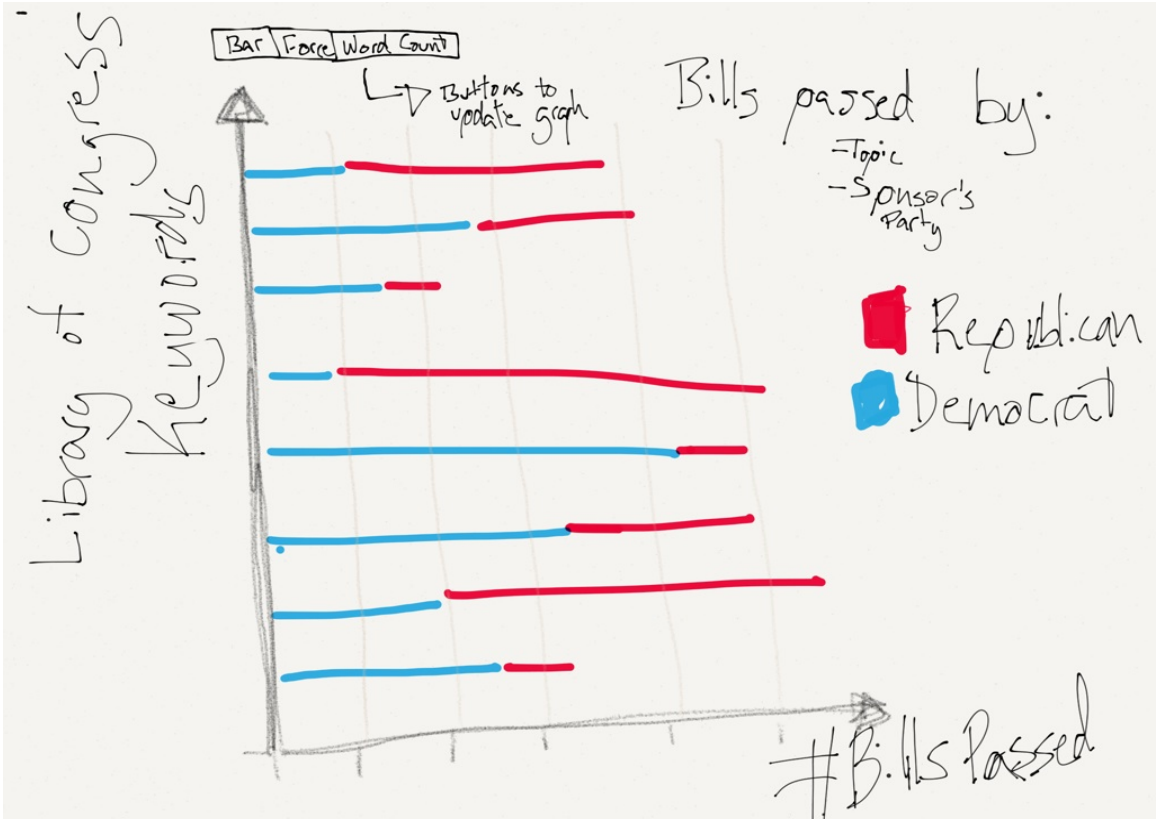
various bill and legislator attributes
- Create different preset filters based on patterns seen during the exploratory analysis

8. Project Schedule. Make sure that you plan your work so that you can avoid a big rush right before the final project deadline, and delegate different modules and responsibilities among your team members. Write this in terms of weekly deadlines.

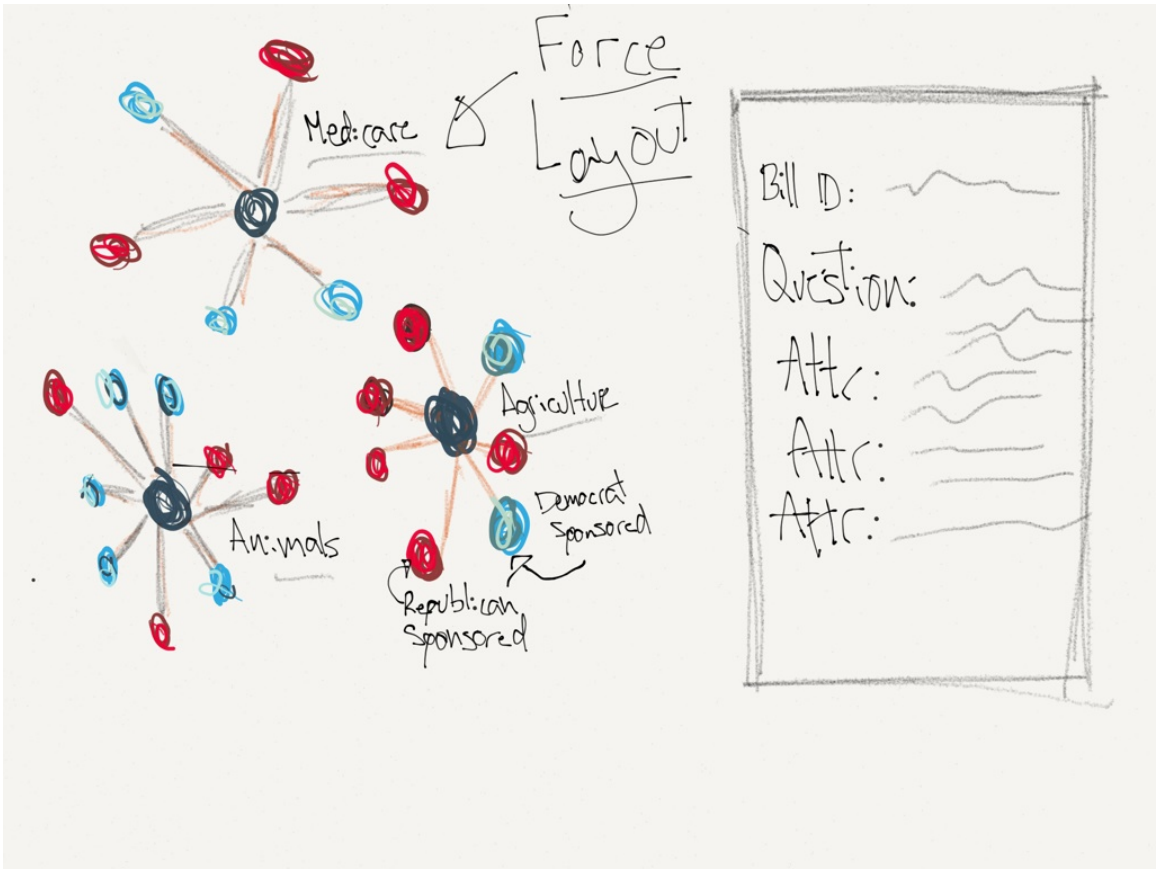
- March 22nd: Have a prototype visualization ready.
- Week of April 14th: Receive feedback from TFs and plan the implementation of the next features to incorporate.
- April 27th: Have the final version of the visualization ready and begin drafting a script for the screencast.
- Week of April 28th: Finish final details of the visualization, documentation, and screencast.
- Lastly, each set of milestones in this schedule should be accompanied by entries into the process book.

Note: In problem 5, we were told to supply an image of what our visualization could look like. After much consideration, we decided to change it to the following:

Visual 1 – Overview Visual



Visual 2 and 3: Table of force layouts and detailed view of Bill
(Each force layout is a classification and each node is a bill)
(When a node is clicked, the detailed layout will appear on the right with the appropriate bill information)



GIT Repository Now Set Up for Group Collaboration

I have set up a private GitHub repository (https://github.com/yashspatel23/cs171-final-project) for my group so that everyone can share their contribution code easily rather than having code mailed around. Moreover, I have shared this repository with cs171tf so that they will be able to pull and test our code.

Sunlight Foundation API

Needing an API to gather information for the Federal Legislation Bills, we decided as a group that Sunlight Foundations Congress API would suffice and work well for our visualization. The API can be found at https://sunlightlabs.github.io/congress/.
Some information about the API:

A live JSON API for the people and work of Congress, provided by the Sunlight Foundation.

- Look up members of Congress by location or by zip code.
- Official Twitter, YouTube, and Facebook accounts.
- The daily work of Congress: bills, amendments, and nominations.
- The live activity of Congress: past and future votes, floor activity, hearings.

All requests require a Sunlight API key. An API key is free, and has no limits or use restrictions.

Some information about methods to gather information:

Path	Description
/legislators	Current legislators' names, IDs, biography, and social media.
/legislators/locate	Find representatives and senators for a `latitude/longitude` or `zip`.
/districts/locate	Find congressional districts for a `latitude/longitude` or `zip`.
/committees	Current committees, subcommittees, and their membership.
/bills	Legislation in the House and Senate, back to 2009. Updated daily.
/bills/search	Full text search over legislation.
/amendments	Amendments in the House and Senate, back to 2009. Updated daily.
/nominations	Presidential nominations before the Senate, back to 2009. Updated daily.
/votes	Roll call votes in Congress, back to 2009. Updated within minutes of votes.
/floor_updates	To-the-minute updates from the floor of the House and Senate.
/hearings	Committee hearings in Congress. Updated as hearings are announced.

/upcoming_bills	Bills scheduled for debate in the future, as announced by party leadership.

The following is a sample result of vote numbers and the Bills result:


```
"results": [
  {
    "breakdown": {
      "total": {
        "Yea": 222,
        "Nay": 190,
        "Not Voting": 19,
        "Present": 0
      }
    },
    "result": "Passed",
    "roll_id": "h43-2013"
  },
  {
    "breakdown": {
      "total": {
        "Yea": 261,
        "Nay": 154,
        "Not Voting": 16,
        "Present": 0
      }
    },
    "result": "Passed",
    "roll_id": "h44-2013"
  }
  ...
]
```

Constructing the Data Structure

In order to get to get our data structure working, we had to learn how to pull information from the API. In doing so, we wrote some notes for others to understand how we created our data structure**:**

- apikey = 2282ac571e0b46b69bb7879f4de9b158

1. There is no sentiment analysis. Despite classifying bills by different keywords there is no visibility into the goal of the bill
    - To counter this it could be useful to display a column chart that shows how many times D vs R sponsored each type of bill.
        - It might be unnecessary to go beyond showing simply D and R regardless of chamber. Yet chamber could be incorporated by having two columns per party: one column for each chamber and one group for each party
2. Upon highlighting a bar, a pie chart could appear that shows the number of bills passed vs failed for this query.
    - This could also be a vertical bar graph to the right
    - There could also be a second chart or graph that shows how many bills of this type were enacted vs vetoed
        - This could be efficiently displayed by using a stacked column chart where one column shows bills rejected and the other column is a stacked column that is composed of bills enacted and bills vetoed (with vetoed bills at the bottom?)
3. Bills:
    - Top Keywords: http://thomas.loc.gov/cgi-bin/bssQuery/?&Db=113&srch=/home/LegislativeData.php?n=BSS&Opt=f
    - List:
        - Agriculture and food, Animals, Armed forces and national security, "Arts, culture, religion", "Civil rights and liberties, minority issues", Commerce, Congress, Crime and law enforcement, Economics and public finance, Education, Emergency management, Energy, Environmental protection, Families, Finance and financial sector, Foreign trade and international finance, Government operations and politics, Health, Housing and community development, Immigration, International affairs, Labor and employment, Law, Native

Americans, Private legislation, Public lands and natural resources, "Science, technology, communications", Social sciences and history, Social security and elderly assistance, Social welfare, Taxation, Transportation and public works, Water resources development

Working query for vote breakdowns:

http://congress.api.sunlightfoundation.com/votes?vote_type=passage&result=Passed&fields=voted_at,chamber,bill_id,question,result,breakdown.party&apikey=2282ac571e0b46b69bb7879f4de9b158&per_page=50&page=1

1. Filters:
   - Vote Type = Passage
   - Result = Passed
2. Fields:
   - Chamber
   - Bill ID
   - Question
   - Result
   - Breakdown by party
   - Page 1

Needs to iterate through:
http://congress.api.sunlightfoundation.com/votes?vote_type=passage&fields=voted_at,chamber,bill_id,question,result,breakdown.party&apikey=2282ac571e0b46b69bb7879f4de9b158&per_page=50&page=1&query=

Working query for bills:

http://congress.api.sunlightfoundation.com/bills?actions.result=pass&fields=short_title,bill_id,history.vetoed,history.enacted&apikey=2282ac571e0b46b69bb7879f4de9b158&per_page=50&page=1&keywords=

1. Filters:
   - Keywords = [keyword]
   - Actions.result = pass
2. Fields:
   - Short title
   - Bill ID

- Was the bill vetoed?
- Was the bill enacted?

Data structure layout:

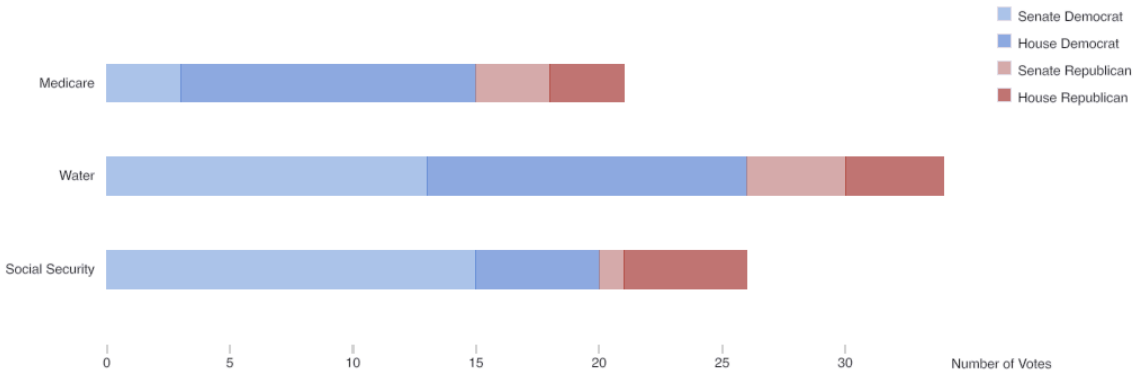
Array

[Keywords]:
  Bill_ID
  Voted at
  Chamber
  Question
  Result
  Breakdown:
    R:
      Yea
    D:
      Yea

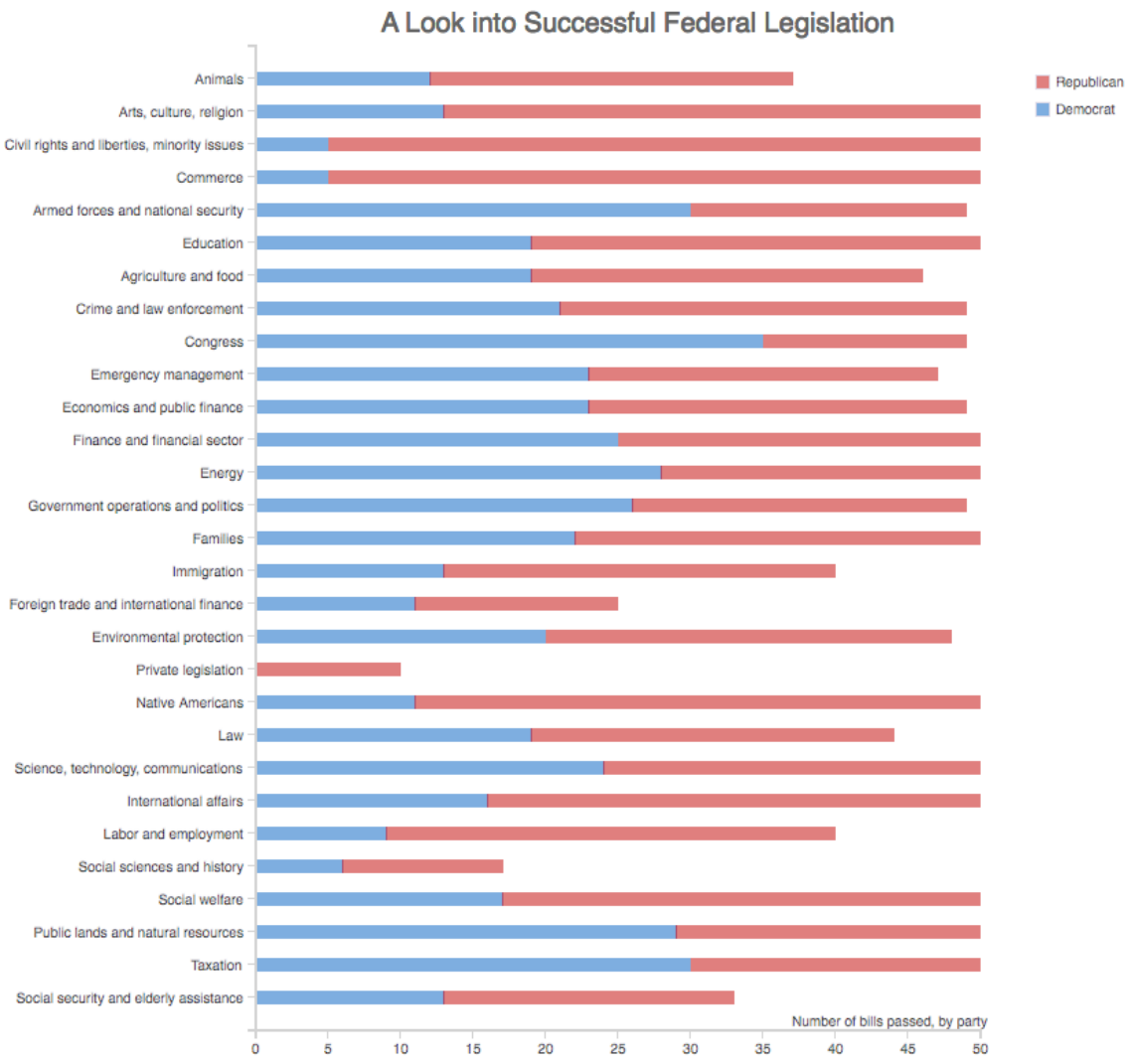
Creating the Main Overview Visualization

After much thinking, our group decided that the best way to present our information would be in a bar graph. The first view, an overview visual, will be displayed in a bar graph that shows approximately 30 classifications (a category to which a bill belongs) and show how many bills passed and which House was responsible. This view will help show viewers what each House is interested in and what classification receives the most attention. For example, if the House of Republican is responsible for 30 bills being passed in Transportation and Public Works, which is the classification with most bills passed, we know that the main focus of Republicans is for said classification and that this is viewed as the most important.

Initially, we began by creating a bar graph with fake local data while our data structure was still being constructed. The visual looked like this:

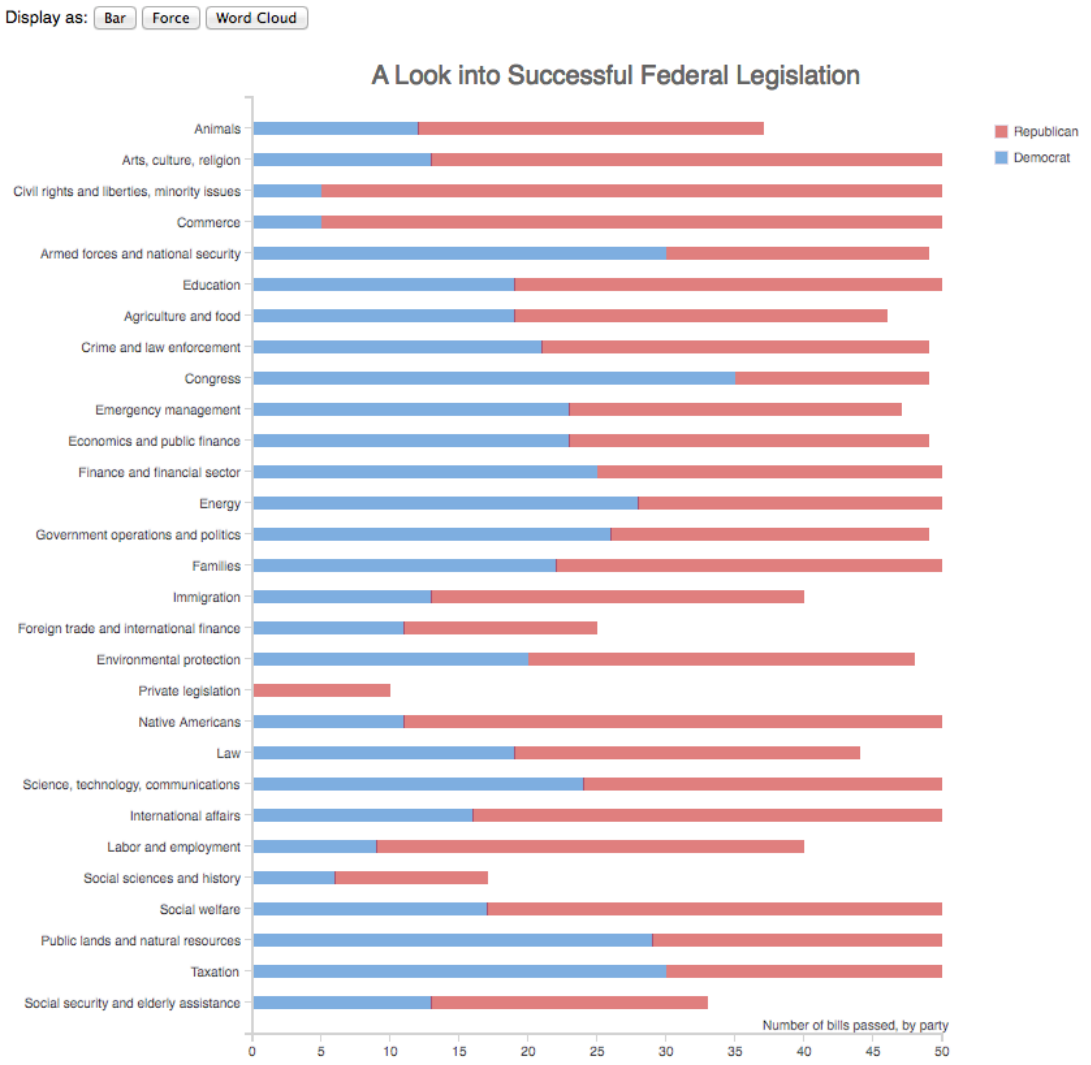


Then, after getting our data structure working, we replaced this false data with real data and the visualization began looking much better:



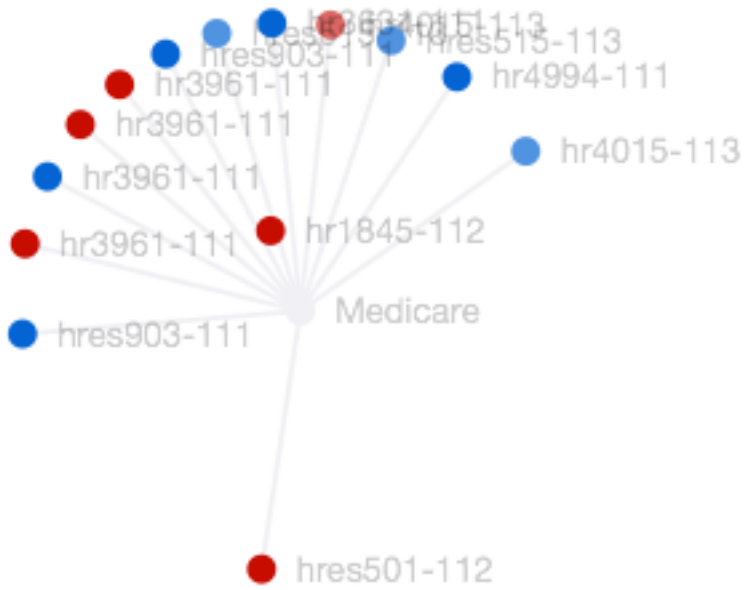
Also, in order to connect the overview layout with the other layouts, we created radio buttons atop the graph:

With this, the main overview visual looks like:



Force Layout and Detailed Visualization

Now, a user wishes to view more specific details about a bill, they have the ability to switch to a force layout by clicking the force button atop the bar graph. This will take them to a visualization that looks something like:



As we can examine, this is a classification (a category to which a bill belongs) force layout. For each classification, each node represents a bill. As a user, we can click on a node and to the right of this visualization, a box with specific details will appear:

Vote question:
On Motion to Suspend the Rules and Pass, as Amended -- H.R. 1845 -- To provide for a study on issues relating to access to intravenous immune globulin (IVIG) for Medicare beneficiaries in all care settings and a demonstration project to examine the benefits of providing coverage and payment for items and services necessary to administer IVIG in the home

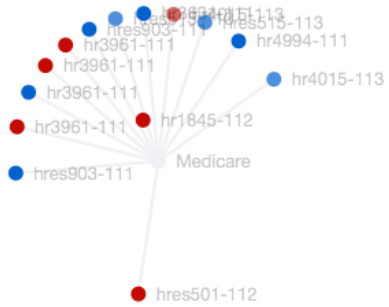
Vote ID:
hr1845-112

Voted Year:
2012

Result:
Passed

Source:
http://clerk.house.gov/evs/2012/roll634.xml

The combined visualization is:



Vote question:
On Motion to Suspend the Rules and Pass, as Amended -- H.R. 1845 -- To provide for a study on issues relating to access to intravenous immune globulin (IVIG) for Medicare beneficiaries in all care settings and a demonstration project to examine the benefits of providing coverage and payment for items and services necessary to administer IVIG in the home

Vote ID:
hr1845-112

Voted Year:
2012

Result:
Passed

Source:
http://clerk.house.gov/evs/2012/roll634.xml

In order to cover all the classifications, we intend to create a table of said force layouts, one for each classification. Then the user will be able to view each bill and its importance. Moreover, we intend to adjust the nodes in size to the House who is responsible for the bill passing. These customizations will happen really soon.

TF Meeting Synopsis and Remaining Tasks

We had a Google Hangout with our TF and present our progress and future implementation plans. She was really impressed with our progress. She recommended some additions to our project that would make it better:

1) Status Loading Bar - Perhaps check out Twitter Bootstrap

2) Is the stacked graph for all Republican and Democrats bills in Senate and House? You can simply clarify in label if you don't have a chance to divide the stacks into 4 categories (House-D, House-R, Senate-R, Senate-D).

3) In stack graph, perhaps have the exact number of bills appear over the bars or a tooltip to help visitor read the graph more easily.

4) Make "Medicare'" title more prominent in force graph

Besides implementing the things that Mimi mentioned, the following are things that we still need to implement:

1) Create a data structure for the detailed layout

2) Modify the word cloud so that word sizes are proportionate to the number of times they appear in bills.

3) Implement the detailed force layout. Also create an epicenter around which the force layout will reside

4) Add tooltips to the bar graph so that it makes reading the value of each bar easier.

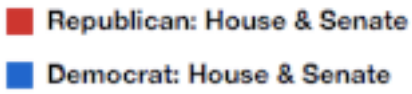
5) Create a better transition selection bar for alternating between the various views.

6) Implement a loading message that offers a loading notice for the word cloud since it takes a while to load

7) Test and fix bugs

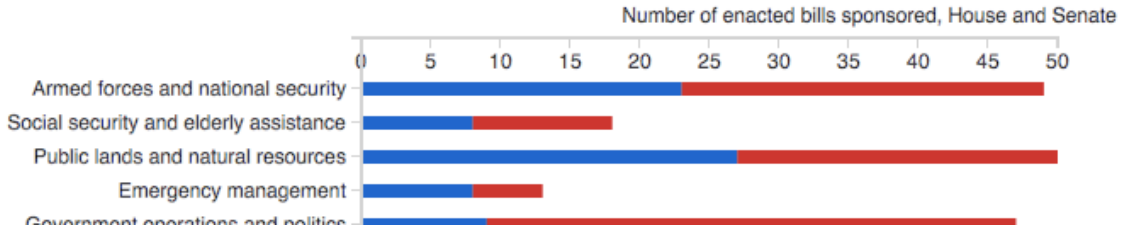
Finalizing Implementation of Bar Graph

There were a few things that needed adjustment before the bar graph implementation was finalized. Firstly, the legend needed clarification that each bar accounted for the total number of bills passed for both the House and the Senate. The implementation now looks like:

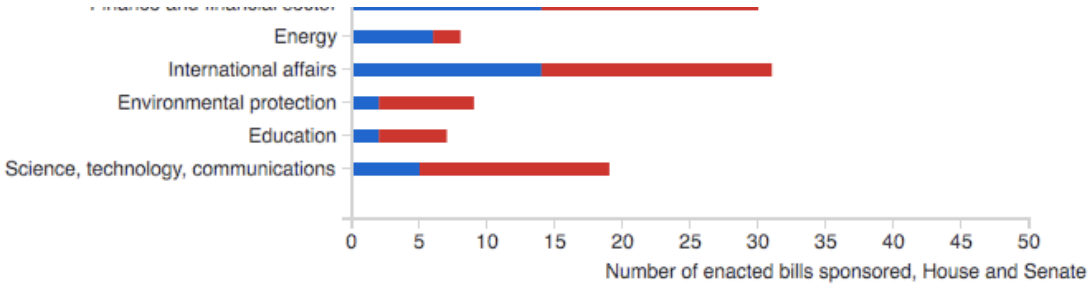


The next change is in regards to the x-axis. Currently, we have placed one x-axis at the bottom of the graph. Due to the vast number of bars, it may be inconvenient to have the user scroll to the bottom each time. Hence, we added an additional axis to the top of the graph. The implementation now looks like:

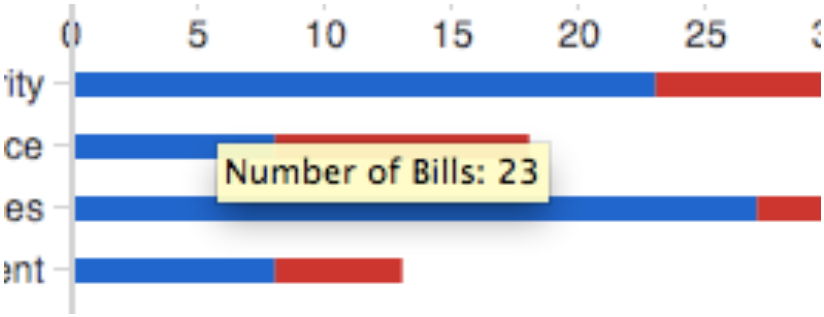
Top x-axis:



Bottom x-axis:



In addition to these changes, it was recommended that we implement a tooltip that would make reading the specific values of each bar easier. Many times, people may have trouble distinguishing between say a 22 and 23. Hence a tooltip was implemented using information from the data structure. To view the tooltip, a user will scroll over a bar and wait 1 second. After implementation, this is what it looked like:



With these changes, the bar graph was finalized. No additional changes are required at this point in time.

Updating our Data Structure for Detailed View

We needed to adjust our data structure for the detailed view. Currently the data structure did not pull any information from the API to store information such as Bill ID, Bill Question, Keyword, Voted Year, or Sponsor Party.

Hence, to pull specific information, we had to adjust our python script as follows:

```
def getBillData(keyw):
    bills_params = {
                    'apikey': '2282ac571e0b46b69bb7879f4de9b158',
                    'keywords': keyw,
                    'history.enacted': 'true',
                    'per_page': 50,
                    'fields': '''sponsor.party,chamber,official_title,summary_short,
                                short_title,popular_title,bill_id,history.vetoed,
                                history.enacted,last_vote_at,urls.govtrack'''
                    }

    endpoint = 'https://congress.api.sunlightfoundation.com/bills'

    response = requests.get( endpoint, params=bills_params)
    data = response.json()
    data = data['results']

    return data
```

This would return information in a data structure that looked like:


```
"Armed forces and national security": {
  "votes": {
    "I": 0,
    "R": 5216,
    "D": 6330
  },
  "bills": [
    {
      "source": "https://www.govtrack.us/congress/bills/113/hr3304",
      "sponsor": "D",
      "short_summary": "(This measure has not been amended since the House agreed to Ser
      "year": "2013",
      "question": "An act to authorize the President to award the Medal of Honor to Benr
      "chamber": "house",
      "bill_id": "hr3304-113",
      "popular_title": null
    },
    {
      "source": "https://www.govtrack.us/congress/bills/113/hr3302",
      "sponsor": "R",
      "short_summary": "Designates the Department of Veterans Affairs (VA) medical cente
      "year": "2013",
      "question": "To name the Department of Veterans Affairs medical center in Bay Pine
      "chamber": "house",
      "bill_id": "hr3302-113",
      "popular_title": null
    },
    {
      "source": "https://www.govtrack.us/congress/bills/113/hjres91",
      "sponsor": "R",
      "short_summary": "Department of Defense Survivor Benefits Continuing Appropriatior
      "year": "2013",
      "question": "Making continuing appropriations for death gratuities and related sur
      "chamber": "house"
```

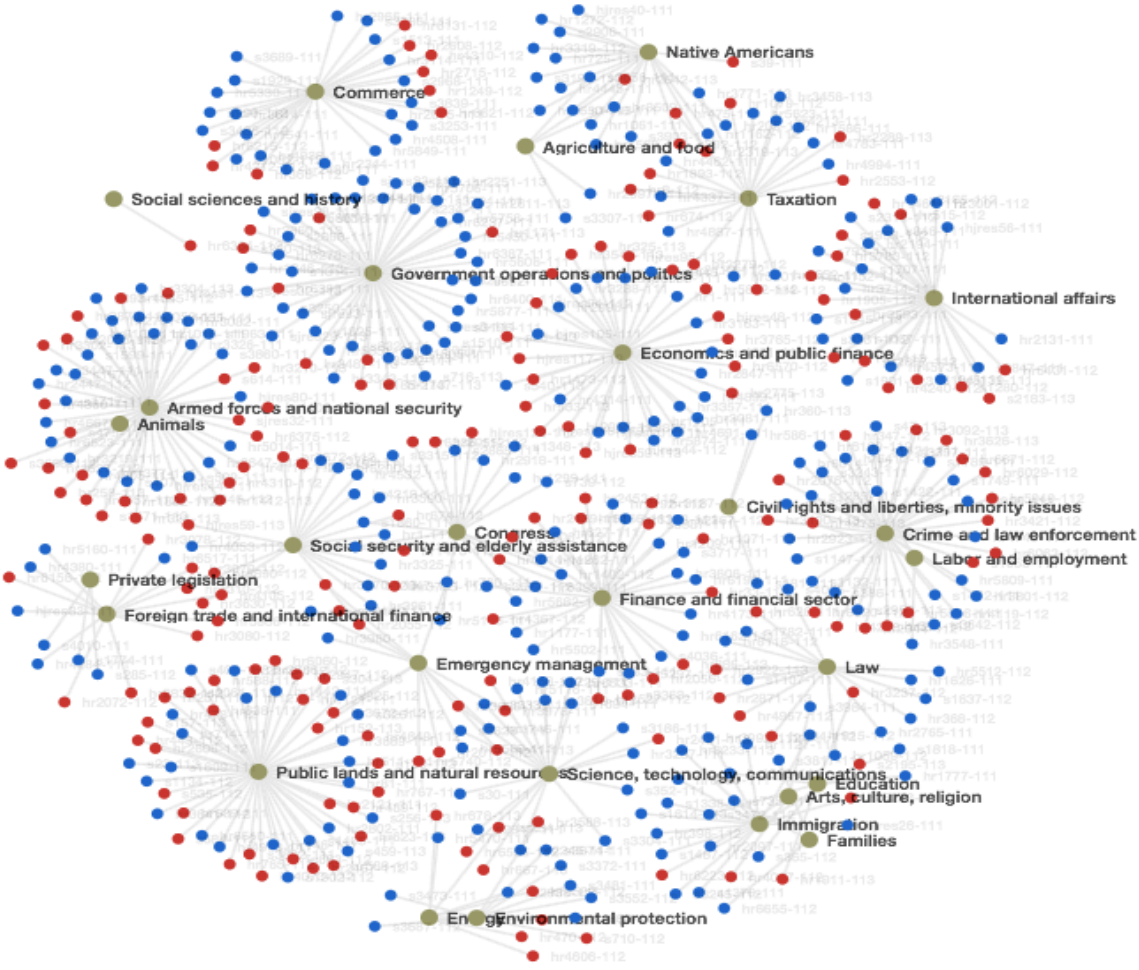

With this updated data structure, we were now ready to implement the force layout and detailed view.

Implementation of Force Layout and Detailed View

Currently in regards to the force layout, we were able to implement a single force layout that would use fake data to create nodes around a central classification. Also, when the node was clicked, a detailed view would appear to the right of the force layout and present information from fake data.

Now that real data is available in our data structure, we needed to decide on a clever way to implement all the force layouts. After much discussion, we decided that we would create an epicenter around which all the force layouts would be centered. That being said, we began to implement.

After finishing implementation, this was the result:



Now that the force layout was implemented, we needed start implementing the detailed view. We liked how the existing detailed view looked and so we need to change a few things around so that it fetched real data from our data structure. The result was:

BILL DETAIL

Bill Id:
s3186-111

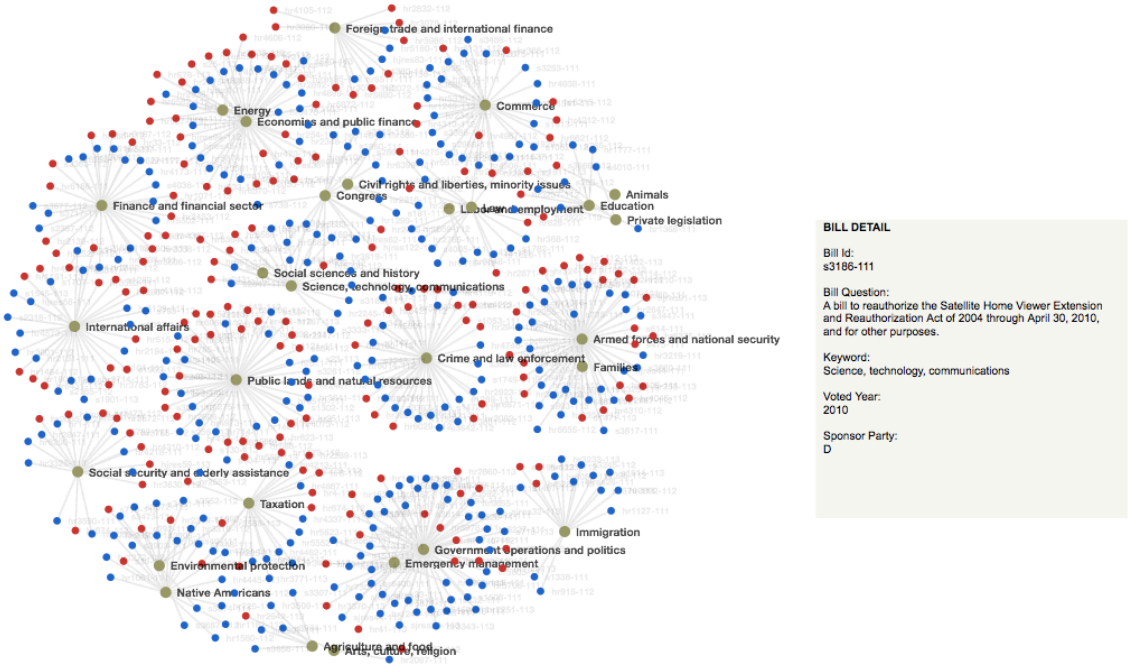
Bill Question:
A bill to reauthorize the Satellite Home Viewer Extension and Reauthorization Act of 2004 through April 30, 2010, and for other purposes.

Keyword:
Science, technology, communications

Voted Year:
2010

Sponsor Party:
D

The force layout and detailed view now looks like:



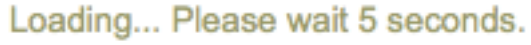
Implementation of Word Cloud

The word cloud we currently have is constructed from fake data and does not adjust size based on the number of times the word appears. Hence, in order to construct a word cloud that was unique, we planned to implement using information that our data structure already.

The implemented result was:

Moreover, when a user visited the word cloud layout, it would take a few seconds for the word cloud to display. Hence, we implemented a message that appeared when a user visited the word cloud and disappeared as soon as the word cloud was displayed.

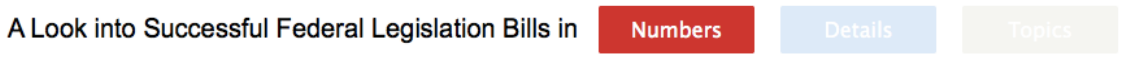
This is what the message looked like:



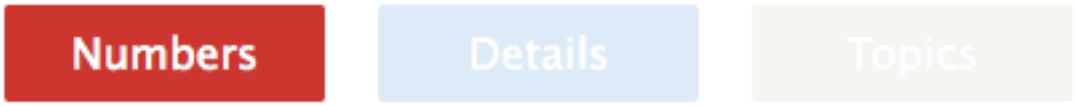
Transitioning Between Visual Layouts

Now that all the views are complete, there was one last thing that we wanted to modify: the transitioning navigation buttons between visual layouts. Currently, we were using simple buttons that worked, but didn't look nice. To change this, we modified their size, color, and position.

After implementation, they looked like:



In more detail:



This allowed users a fixed navigation between views and made transitioning between visual layouts VERY simple.

This was our final touch to the project. We now believe the project to be complete.

Concluding Remarks

Logistics

The work of this visualization was split evenly amongst all team members. Each individual contributed in different areas and gained exposure to all aspects of the visualization. At some point or another, each individual had the opportunity to make contributions to each part of the visualization. Although we were not able to meet locally and work together, we were VERY efficient by constantly emailing our progress, problems, solutions, and ideas to each other. Moreover, we had organized weekly Google Hangout meetings to discuss our current position on the project. We would plan what needed to be complete, alternative ideas to visual layouts, potential problems, and work distribution during these Hangouts. Afterwards, throughout the week we would communicate via emails, phone calls, and text messages. All in all, we worked VERY well as a team and were diligent and helpful throughout the implementation process.

Conclusion

We hope that you will enjoy this visualization as much as we enjoyed implementing it. The process of implementing such visualization was great experience for all of us. We received first hand exposure to how much work goes behind such a beautiful, integrated visualization that we sometimes take for granted. Moreover, each individual gained exposure to all aspects of the project. Each team member made contributions to all parts of the project. There were many issues that we came across, many things that we disagreed upon as a team, and faced challenges of having varying working schedules. However, we dealt with these problems VERY efficiently and managed to help each other in times of trouble. In conclusion, this project was a great experience and exposed us to the world of data visualization!