Final Project Report

Digital Democracy Organization Disambiguation Clustering

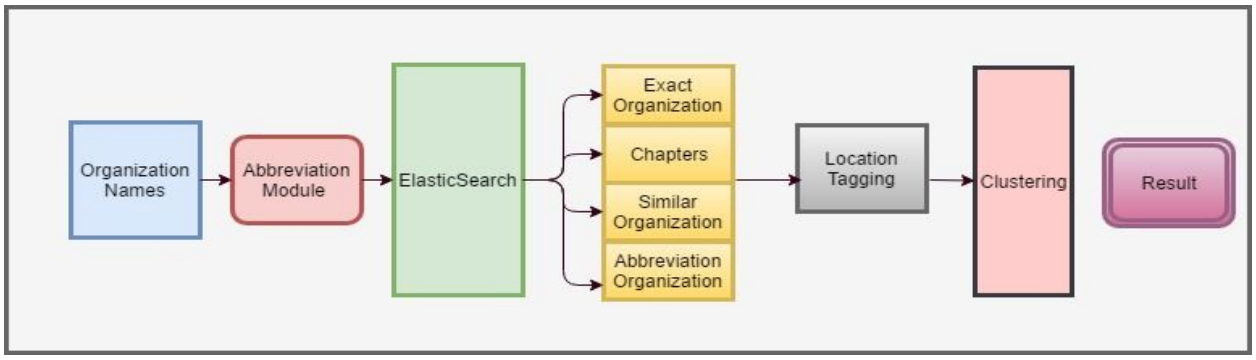
Aditya Budhwar, Brandon Cooper, James Caudill

1 Introduction

Digital Democracy is a organization that gathers and publicizes california state government proceedings. The data it holds are transcribed hearings, testimonies and floor discussions on anything from proposed bills to tax cuts. Since having all that data alone doesn't help much, they also learn from the data through NLP and machine learning. When gathering such a large general corpora that will eventually be searched for specific people, bills, or businesses, some basic word disambiguation is necessary. When a user wants to know about all bills sponsored by Cal Poly SLO the search algorithm needs to return all documents related to that specific institution. But not everybody says "Cal Poly SLO", there are many variations: California Polytechnic State University, Cal Poly, Cal State SLO, etc. Rather than building in the disambiguation into the search engine which needs to operate quickly, there should already be an index of companies with multiple names. To create such index, an unsupervised clustering algorithm with custom distance metric sorts all the transcribed organizations into likeness groups. Then the search engine can use all connected words to increase the algorithms recall, something general search engines usually lack. Our group set out to create the custom distance metric and calculate both the number of organization clusters and all associations.

2 Initial System Design

Our initial system design involved running each organization mention through ElasticSearch in order to retrieve a shortlist of related organizations to cluster together. In the end, our ElasticSearch search engine and clustering process became separate systems, with ElasticSearch searching through the base organization TSV file, and the clustering system clustering mentions directly from the same organization TSV file.



3 Development

In the process of developing the solution we started with exploring the various clustering approaches such as KMeans, Agglomerative clustering, DBScan, Affinity Propagation. The issues we faced with the clustering approaches in general was that they were very slow and due to large number of organization names the accuracy of the clusters was very low moreover noise added to the bad clustering results as well. We also explored Levenshtein distance methodology to find the distance between two words and distance at character level, the problem with this approach was the accuracy here organization names like

Adobe systems and Perrot Systems also match. Post these hiccups we decided to divide our project into two segments one based on ElasticSearch and other based on clustering.

Elasticsearch

In elasticsearch we indexed organization name, ID, and abbreviation of the organization name. To calculate the abbreviation we used technique where we found multiple abbreviation forms once for entire organization name and other by removing stop words such as conjuncts. Post indexing data for all the entries the search was divided into 4 sections:

1. Exact Organization Matchrm
2. Chapters Organization
3. Similar Organization
4. Abbreviation match Organization

The results of these categories are then processed for named entity recognition for tagging the location. The future plan here is to use set generated by elasticsearch and using location tagging as a way to find chapters and then clustering the results. The demo of Elasticsearch is hosted on frank server on www.frank.ored.calpoly.edu/MDSearch/index_organization.html

Clustering

In the end, we settled on SciKit-Learn's implementation of the DBScan clustering algorithm, which infers the number of clusters from the data, and allowed us to tune the minimum cluster size, as well as the maximum distance from centroid points to other organizations within the cluster. Since many organizations have only a single mention, we set the minimum cluster size to 1 to allow single organizations to exist as separate single-node clusters. In order to cluster the organizations, we first preprocess the mentions to separate any locations in the mention from the organization name using a combination of SpaCy's named entity recognition system and a list of U.S. states and their abbreviations. Next, our algorithm splits organization qualifiers (e.g. LLC., Inc., co., etc.) from the mention, leaving just the base organization name. After this, we query for an organization URI (uniform resource identifier) from a locally running DBPedia server using the remaining text of the organization mention. If DBPedia cannot match a URI to the organization, we leave the URI empty for the current organization. Finally, we compute the distance between each organization according to our distance algorithm outlined below. With these precomputed distances, we fit a DBScan clustering model to the resulting matrix of distances, with a maximum distance from centroid points of 40.0.

Organization Mention Distance

For each organization mention, M_i, let O_i be the preprocessed organization mention with location and organization qualifiers removed from M_i. Let Q_i be the list of organization qualifiers (Inc., LLC, co., etc.) found in M_i, let U_i be M_i's URI found from DBPedia, and let L_i be the locations parsed from M_i. Let cosineSimilarity be a function that computes the average cosine similarity of the word vectors (from a model pre-trained on the English language) of two organization names. Let wordDistance be another function which computes the number of words which differ between two lists of words.

To compare two organization mentions M_i and M_j:

If U_i *and* U_j *are not empty* :

$$Distance = 100 * (1 - cosineSimilarity(U_i, U_j))$$

Else :

$$Distance = 100 * (1 - cosineSimilarity(O_i, O_j))$$

$$Distance = Distance + 2 * wordDistance(L_i, L_j) + wordDistance(Q_i, Q_j)$$

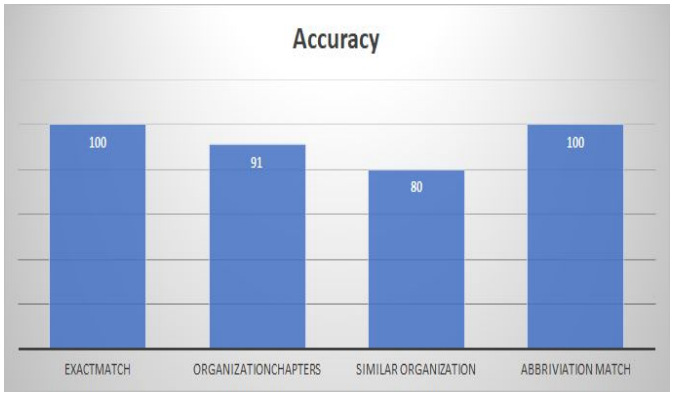
Visualization

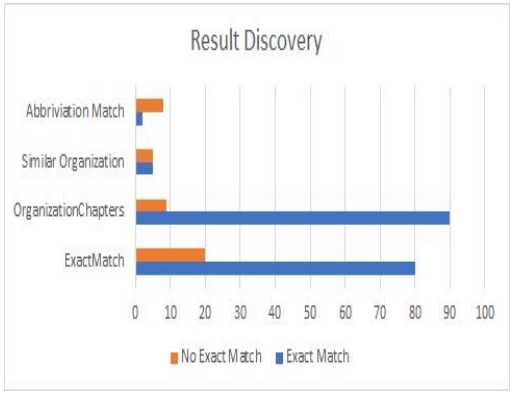
The final visualization part of the project was created with the help of a javascript library for networks called Vis.js. The tool allows for a large number of clusters to be shown and interacted with. It is located on our bootstrapped webpage. This web page is a single scrolling page with a title, the clustered networks, the division of labor, links to this report, the presentation slides and our code repository. This link is accessible to the world and shows clusters based off of the last run of our clustering algorithm. There is already a lot of data for the javascript to load, making the page load time slow, there is no way we can cluster every request. There wouldn't really be a need to do so either unless we were adding more organizations. This might be possible, and therefore could be considered as future work for another group. The main challenge of using this library to show our clusters is that the physics simulation can only handle around 600 nodes without stalling the browser. So, instead of showcasing all of our clusters, we chose the top 120 with the most linked organizations. The link to our website is: http://frank.ored.calpoly.edu/~jacaudil/www/index.html

4 Results

ElasticSearch

The ElasticSearch results have been tabulated on random 10 entries below.





Clustering

When examining a sample of the clustered organization results, we found that our clustering algorithm grouped organizations together mostly by keywords in common between the mentions. This resulted in clusters of organizations from related fields, such as law offices, transportation services, and high schools, rather than organization co-references. For example, the main law office cluster consisted of "Law Offices of Rusty Selix", "Law Office of Roger Koll", "Law Offices of Fei Pang", and several other law offices.

5 Conclusions

We found that our baseline clustering algorithm mostly clustered organizations around the type of organization, such as law office, high school, or political action committee, rather than grouping the same physical organizations to their various mentions. Though we couldn't implement the entire initial system design we had planned, but we believe if we use the results from ElasticSearch to create a shortlist of related organizations to run through our clustering algorithm to produce more accurate clusters.