AUTOMATED CONSTRUCTION & ANALYSIS OF POLITICAL NETWORKS VIA OPEN GOVERNMENT & MEDIA SOURCES

DIEGO GARCIA-OLANO
MARTA ARIAS
JOSEP LLUÍS LARRIBA PEY

for ECML PKDD 2016 So Good Workshop
THEY MYSTERY THAT IS TEXAS

Population: 27 Million People
THE MYSTERY THAT IS TEXAS POLITICS

Population: 27 Million People

State & Federal Politicians: 250

composed of:
- State Congress: 150 Representatives & 31 State Senators
- U.S. Congress: 36 Representatives & 2 Senators
- State Executive officials:
  Governor, Lieutenant Governor, Speaker of the House,
  Attorney General, Comissioners, etc
- State Judges: 9 Supreme Court & 9 Court of Criminal Appeals
THE MYSTERY THAT IS TEXAS VOTER TURNOUT

Population: 27 Million People

State & Federal Politicians: 250

composed of:
- State Congress: 150 Representatives & 31 State Senators
- U.S. Congress: 36 Representatives & 2 Senators
- State Executive officials:
  Governor, Lieutenant Governor, Speaker of the House,
  Attorney General, Comissioners, etc
- State Judges: 9 Supreme Court & 9 Court of Criminal Appeals

Voter Turnout: 20% for 2015 Governor election.

Texas House Congressional Districts Map. 150 districts total
http://www.whoyouelect.com/texas/texas-house-map.html
1. list of Politicians
2. list of relevant news sites

Construct & Display the networks around these politicians

Case study:
1. 246 currently active Texas elected officials
2. 6 news sites that cover Texas politics
   * the Austin American Statesman, the Dallas Morning News, the Houston Chronicle,
   * the New York Times, the Texas Observer and the Texas Tribune
PRESENTATION OUTLINE

- Introduction
- Overview of WhoYouElect.com
- Automated Construction of Networks & Prior Work
  1. Ego Network of a Politician
  2. Extended Network of a Politician
  3. Automated Summarization of Communities via Topic Modeling
- Conclusions & Future Work

Case Study Results, Networks, Maps, Code: http://whoyouelect.com/texas
Slides: http://www.whoyouelect.com/texas/ecmlpkdd-sogood/
OVERVIEW OF WHOYOUELECT.COM

http://www.whoyouelect.com/texas
WHO IS EDDIE RODRIGUEZ?

Table Of Contents View:
1. EGO "INNER" NETWORK OF A POLITICIAN


or just click on "Inner Network" from the Table of Contents view

**URL parameters**

<table>
<thead>
<tr>
<th>s</th>
<th>name of Politician</th>
</tr>
</thead>
<tbody>
<tr>
<td>show</td>
<td>only show edges with weight greater than or equal to show</td>
</tr>
<tr>
<td>dm</td>
<td>(optional) which distance metric to use. possible values: ss, sn, all, comb. corresponding to Same Sentence, Same or Near, All Co-Ocurrences, Proposed Combined Metric. Defaults to &quot;all&quot;</td>
</tr>
<tr>
<td>near_co</td>
<td>(optional) Near Sentence Coefficient in calculation for Combined Metric</td>
</tr>
<tr>
<td>same_art_co</td>
<td>(optional) Same Article Coefficient in calculation for Combined Metric</td>
</tr>
<tr>
<td>from</td>
<td>(optional) date from which to include articles found. expected format: YYYY-MM-DD</td>
</tr>
<tr>
<td>to</td>
<td>(optional) end date for inclusion of articles. expected format: YYYY-MM-DD. ex: 2008-07-01</td>
</tr>
<tr>
<td>exclude</td>
<td>(optional) which news sources to exclude. possible values: AAS, DMN, HC, NYT, TXOB, TXTRB. use commas to exclude multiple</td>
</tr>
</tbody>
</table>
Eddie Rodriguez has most same sentence occurrences with

Politician Dawnna Dukes - (17)
Person Hugh Keefe - (6)
Organization Tesla - (8)
Location AUSTIN - (52)
Bill House Bill 506 - (2)
Misc Travis County - (27)

<table>
<thead>
<tr>
<th>Politicians:</th>
<th>Organizations:</th>
<th>Persons:</th>
<th>Locations:</th>
<th>Bills:</th>
<th>Misc:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dawnna Dukes - 17</td>
<td>Tesla - 8</td>
<td>Hugh Keefe - 6</td>
<td>AUSTIN - 52</td>
<td>House Bill</td>
<td>Travis County - 27</td>
</tr>
<tr>
<td>Elliott Naishat - 16</td>
<td>Milford Superior Court - 7</td>
<td>Ciro Rodriguez - 6</td>
<td>East Austin - 10</td>
<td>House Bill</td>
<td>Austin Democrat</td>
</tr>
<tr>
<td>Tom Craddick - 14</td>
<td>Farm to Table Caucus - 5</td>
<td>Richard Raymond - 5</td>
<td>San Antonio - 8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mark Strama - 14</td>
<td>Mexican American Legislative Caucus - 3</td>
<td>Paul Moreno - 4</td>
<td>Texas House - 6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lon Burnam - 12</td>
<td></td>
<td>Lee Leffingwell - 4</td>
<td>Milford - 6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2016 RIVA DEL GARDA
2. EXTENDED VIEW OF A POLITICIAN


* Edges weighted according to the proposed "Combined" metric!
* Placement of nodes/communities is calculated to maximize separation and clarity

or just click on "Extended Network" from the Table of Contents view

URL parameters

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>name of Politician</td>
</tr>
<tr>
<td>t</td>
<td>only show edges with weight greater than or equal to threshold t. Defaults to 15.</td>
</tr>
<tr>
<td>cl</td>
<td>number of communities to discover in network. Defaults to 25.</td>
</tr>
</tbody>
</table>
3. AUTOMATED SUMMARIZATION OF COMMUNITIES

1. Treat the articles of a given community collectively as a single corpus.

2. Run an initial TF-IDF procedure to filter terms and reduce noise, and

3. followed by Latent Dirichlet Allocation (LDA) to derive topics.

**We can weigh articles by their relative importance in the community.
TOPIC MODELING TO SUMMARIZE A SINGLE COMMUNITY

```r
library("topicmodels")
library("tm")
k = 5 #num of communities to look for
highend = 4000 #want less than this many terms
lowend = 2000 #want more than this many terms
tsvfile = "eddie_rodriguez-articles.tsv"
articlesForCommunity <- read.csv("lda/community-articles-lda.csv") #community article texts
colnames(articlesForCommunity) <- c("url", "date", "entities")
ngramtype <- 1
weightbyentity = T
article_cutoff = 1
runTopicModelingOnCommunity(tsvfile,k,highend,lowend,
                             ngramtype,article_cutoff,weightbyentity)
runTopicModelingOnCommunity <- function( ... ){
  # ... load tsv to get article texts and put in JSS_papers list
  corpus <- Corpus(VectorSource(sapply(JSS_papers[, "text"], remove_HTML_markup)))
  JSS_dtm <- DocumentTermMatrix(corpus, control = list(stopwords = TRUE,
                                 minWordLength = 3, removeNumbers = TRUE, removePunctuation = TRUE))
  term_tfidf <- tapply( JSS_dtm$v/row_sums(JSS_dtm)[JSS_dtm$1, JSS_dtm$j, mean],
                        * log2( nDocs(JSS_dtm) / col_sums(JSS_dtm > 0) )
  cutoffvals <- get_cutoffval_and_type(term_tfidf,highend,lowend)
  # ... filter corpus by cutoff vals ...
  jss_TM <- LDA(JSS_dtm, k = k, control = list(seed = SEED))
  Topic <- topics(jss_TM)
  Terms <- terms(jss_TM)
  # ... generate most frequent topics of the articles and the terms for each one
}```
### Entities Info:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Party</th>
<th>Level</th>
<th>Position</th>
<th>District</th>
<th>G Degree</th>
<th>L Degree</th>
<th>G Page Rank</th>
<th>G Translucity</th>
<th>G Strength</th>
<th>L</th>
<th>L Strength</th>
<th>L Betweeness</th>
<th>Articles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Andrew Smiley</td>
<td>PER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>0.5</td>
<td>1766</td>
<td>1766</td>
<td>0</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>AUSTIN</td>
<td>LOC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>34</td>
<td>10</td>
<td>100</td>
<td>0.033</td>
<td>2810.2</td>
<td>1582.6</td>
<td>63</td>
<td>155</td>
<td></td>
</tr>
<tr>
<td>Sustainable Food Center</td>
<td>ORG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>0.5</td>
<td>1421.5</td>
<td>1421.5</td>
<td>0</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>House Bill 1392</td>
<td>BILL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>5</td>
<td>0</td>
<td>0.357</td>
<td>648</td>
<td>504</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Department of State Health Services</td>
<td>ORG</td>
<td></td>
<td>statewide-active</td>
<td></td>
<td></td>
<td>7</td>
<td>5</td>
<td>0</td>
<td>0.357</td>
<td>648</td>
<td>504</td>
<td>0</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>Susan King</td>
<td>POL</td>
<td>Republican</td>
<td>Representative</td>
<td></td>
<td>71</td>
<td>7</td>
<td>5</td>
<td>0</td>
<td>0.357</td>
<td>607.5</td>
<td>463.5</td>
<td>0</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Uncle Billy</td>
<td>PER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>180</td>
<td>180</td>
<td>12</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Kel Seliger</td>
<td>POL</td>
<td>Republican</td>
<td>statewide-active</td>
<td></td>
<td>31</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 5.1 Summarizing a Community Via Topic Modeling with topics composed of single terms

<table>
<thead>
<tr>
<th>Topic</th>
<th>Articles</th>
</tr>
</thead>
<tbody>
<tr>
<td>25.45% tax, strayhorn, rates, students, craddick, car, tesla, gambling, industry, cars</td>
<td></td>
</tr>
<tr>
<td>21.82% food, farmers, maps, markets, caucus, redistricting, doggett, plaintiffs, latino, map</td>
<td></td>
</tr>
<tr>
<td>18.79% craddick, gambling, interest, lenders, loans, loan, rates, tax, incentives, annual</td>
<td></td>
</tr>
<tr>
<td>18.79% energy, program, line, latin, market, sanchez, craddick, jobs, fashion, foreign</td>
<td></td>
</tr>
<tr>
<td>15.15% utility, uber, energy, tesla, rates, dealers, lyft, shoes, electric, stores</td>
<td></td>
</tr>
</tbody>
</table>

### Articles Info:

<table>
<thead>
<tr>
<th>URL</th>
<th>Date</th>
<th>Num Times in Community</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.texastribune.org/2005/01/17/easy-as-pie/">www.texastribune.org/2005/01/17/easy-as-pie/</a></td>
<td>2005-01-17</td>
<td>4</td>
</tr>
<tr>
<td><a href="http://www.texastribune.org/2013/05/10/guest-column-early-look-legislative-partisanship/">www.texastribune.org/2013/05/10/guest-column-early-look-legislative-partisanship/</a></td>
<td>2013-05-10</td>
<td>4</td>
</tr>
<tr>
<td><a href="http://www.texastribune.org/2010/03/01/one-question-remains/">www.texastribune.org/2010/03/01/one-question-remains/</a></td>
<td>2010-03-01</td>
<td>4</td>
</tr>
<tr>
<td><a href="http://www.texastribune.org/2008/04/14/how-it-all-came-out/">www.texastribune.org/2008/04/14/how-it-all-came-out/</a></td>
<td>2008-04-14</td>
<td>3</td>
</tr>
</tbody>
</table>
AUTOMATED CONSTRUCTION OF NETWORKS & PRIOR WORK
PRIOR WORKS

1. Time intensive, hand crafted networks
2. Article lookup systems with an impressive, though limited, breadth of sources
3. Paid-for search engine results, only processing first twenty results and only using text present in Yahoo’s search result listings.

We were unable to find any work that leveraged the publically available search engines present in most news websites.
GENERAL OVERVIEW OF GRAPH CONSTRUCTION PROCESS

3.1 Adding Politicians From Open Government Sources

3.2 Adding News Sources

3.3 Set up for Data Acquisition By Template Modification

3.4 Running and storing the Web Search results for each Active Entity

3.5 Processing article results per Active Entity

For each article found for a politician:
1. Split the article into sentences
2. Run Named Entity Recognition over sentences
3. Disambiguate the entities found
4. Filter entities
5. Check if an article is a candidate listing
6. Discovery of relationships between entities
7. Verification and construction of relationships
8. Save Intermediate Results
9. Save Article Metrics

Save Global Metrics and Global Results file

3.6 Generate Individual star and extended views for each Active Entity

http://openstates.org/api/v1/legislators/?state=tx&active=true
https://www.govtrack.us/api/v2/role?current=true&state=TX
http://www.sos.state.tx.us/elections/voter/elected.shtml
GENERAL OVERVIEW OF GRAPH CONSTRUCTION PROCESS

Languages, Libraries, and Databases
Python: general backend work
MongoDB: to store article texts and entity information
MITIE: MIT open source Named Entity Recognition tool
BeautifulSoup and Selenium libraries:
  python web scrapers used in obtaining articles.
  BS4 is for static web pages,
  while Selenium, using the PhantomJs webdriver,
  handles pages constructed by javascript dynamically
langdetect:
  open source python library for language detection
D3.js: network visualizations & maps
jQuery UI: some frontend interactivity functionality
jLouvain: javascript Louvain community detection
html5 webworkers:
  asynchronous, nonblocking JS load of data for graphs

http://openstates.org/api/v1/legislators/?state=tx&active=true
https://www.govtrack.us/api/v2/role?current=true&state=TX
http://www.sos.state.tx.us/elections/voter/elected.shtml
3.1 ADDING POLITICIANS FROM OPEN GOV. SOURCES

- 1. Texas Congress data was obtained from OpenStates.org for both active and inactive members
- 2. Federal Congress data was obtained from GovTrack.us for current federal representatives
- 3. Other Texas state officials data was obtained from the Secretary of State of Texas website via a script

This gives us the metadata for all the politicians
3.2 ADDING NEWS SOURCES

- Dallas Morning News
- Houston Chronicle
- Austin American Statesman
- Texas Observer
- Texas Tribune
- New York Times

A reasonable mix of representative media sources on Texas politics.
3.3 SETUP FOR DATA ACQUISITION BY TEMPLATE MODIFICATION

Two template "web scraper" solutions are provider.

Based on whether a news site renders its site content **statically** or **dynamically** via Javascript.

1. The **static version, based on BeautifulSoup**
   - acquires data more quickly,
   - but can not handle dynamic content.

2. The **dynamic version, based on Selenium/PhantomJS**
   - can handle static or dynamic content,
   - but goes slower than the static solution;

Future work is planned to unify the approach into one template.
3.4 RUNNING AND STORING
THE WEB SEARCH RESULTS FOR EACH ACTIVE ENTITY

This step calls the webscraper template for a politician.
For each news source
- download the list of article urls returned from its internal search engine for that politician
- download full articles into JSON files
- do language detection for the text of the article before importing JSON into MongoDB.

3.5 PROCESSING ARTICLE RESULTS PER ACTIVE ENTITY

Take all the articles downloaded for a given politician, process and store them, and construct graphs for the politician:
1. the Ego "Inner" Network, and
2. the Extended Network View
3.5 PROCESSING ARTICLE RESULTS ... CONTINUED

Go through all the articles for a politician one by one and

1. filter out empty articles, sports articles, and articles not containing the politician's name explicitly.

2. Split article into sentences, and run the MITIE Named Entity Recognition library over each one. This finds "entities" in each sentence and gives each a tag of "person","location","organization" or "misc". Additionally check if "person" tags are "politicians" using our entities DB or whether any Congressional "bill"s exist in the sentence using a heuristic.

3. Run coreference resolution over all the entities found to get an additional dictionary of all distinct entities found in the article.

4. From this and the tagged sentences, we then find and store all co-occurrences that occurred within the same sentence, within three sentences, or outside of that distance for all entities in the dictionary.

5. At this point, the article has been processed and we merge and save it locally.

3.6 CREATE NETWORK VIEWS

Using saved result objects and statistics, construct the Ego and Expanded network data files.
CONCLUSIONS & FUTURE WORK
CONTRIBUTIONS

We presented ...
1. a tool that generates real world political networks from user provided lists of politicians and news sites.
2. enriched with data obtained from open sources to provide structure via verified politician meta-data.
3. the Ego "Inner" and "Extended" graph visualizations
4. a “Combined” distance metric to better assess the strength of relationships between actors in a graph.
5. a proof-of-concept use of topic modeling for labeling communities in a politician’s “extended” network (not shown)

Uses:
- voter education
- creating real world networks for study
- discovering potential news stories
FUTURE WORK

1. an extensive statistical study of the merged "extended" graphs obtained
2. incorporation of city & local APIs for better resolution of elected officials, campaign funding APIs for influence tracking, congressional bill APIs, public health, socio-economic, and voting history APIs
3. all articles aren't equal, and as such weighing article relations differently is very important.
4. better disambiguation of entities, use of alias lists, automated merging tools
5. simplification of web scraping solution & refactoring code to handle "parties" more generally for non-US cases
6. expanding NER solution to provide for more language handling (Catalan for instance)
7. refactoring text-snippet solution for better scalability.
8. developing mechanism for downloading, processing and adding new articles for existing politicians.
9. assessing use of multiplex paradigm by introducing additional link types (“neighboring districts”, “author of bill”, “member of committee”, etc.) for more robust network analysis.
10. leverage posteriors of LDA for better topic analysis, and similarly leverage stochastic community detection methods
11. relationship labeling/role discovery incorporation (signed positive/negative edges when applicable)
12. temporarl community detection work/view
THANKS!

QUESTIONS?
TELL YOUR TEXAS FRIENDS :)