Correlating Language Model Clusters of News Agencies and Political Campaigns with Political Biases

CS224N Final Project - Winter 2011 - Yaron Friedman, Issao Fujiwara

Abstract

Our objective is to attempt to understand how high-level features such as political affiliation and orientation is expressed in low-level statistical features such as 1,2,3-gram language models. We crawl and scrape English text from articles from eight major news agency's sites and nine politician's campaign sites. We train language models with each corpus, and then evaluate each other corpus with these models. We apply clustering techniques to the resulting data, analyzing the correlation of these clustered results with political affiliation and orientation. We observed interesting features in this data, such as the consistent clustering of the obama and biden corpora, as well as a significant split of the candidates cluster by party affiliation. Finally we define a method for extracting relevant words in distinguishing the different corpora that provides us some intuition about the results observed.

Data Collection

Our goal here was to collect text data from different news sources and politicians' campaign material. We were unable to locate good stock sources for these, so we decided to fetch the data ourselves by extracting English text from relevant websites. This process turned out to be significantly harder than we originally thought, but with lots of heuristics we were able to get a decent amount of high quality English text from most sources that we looked at. It's worth noting that we weren't able to necessary gather complete texts, but we have high confidence that the text we gathered is primarily article content.

For the crawling component of obtaining the data, we investigated using many different free web crawlers that are publicly available. Among the few crawlers that we investigated, we spent the most with Heritrix [1], the crawler used by the Internet Archive, and WebSPHINX [2] developed at CMU. While both of these are powerful and sophisticated crawlers, it was actually quite hard to get them to reliably achieve our goal of reliably obtaining as many HTML pages served from a given domain as possible. Surprisingly to us, we ended up having the best luck with the recursive download function of `wget` [3], a widely available UNIX command-line tool:

```
wget -r -l inf -D <domain> <target> -R w  -o log-<source>
```

With that strategy, we were able to reliably download large HTML corpora for each of the news agency and presidential candidate websites that we were interested in.

Once we obtained the raw HTML data for each corpus, we set out to extract the relevant English text from each of the corpora. Our strategy involved parsing the HTML for each page using the BeautifulSoup [4] library and iterating over several heuristics in order to remove as much non-English text as possible. We first stripped out any text that was in a <script>, <style>, <head>, <meta> or <option> tag as well as HTML comments. From there we extracted all text that was a descendant of a <p> and flattening all the text nodes in its sub-trees. By itself, that strategy was enough to get us most of the way towards having only English text in our data, but there was still a lot of extraneous content in that data that made bulding language models from it hard. Some examples included lots of navigational text, copyright notices, other meta English text such as notices about browser compatibility and error pages, and finally ads, comments and related tweets and Facebook messages.

We applied a few more heuristics in order to gather a corpus that was clean enough to use for building language models. We started with a few rules, evaluated the resulting output, and iteratively refined it until we had text that appeared to be strictly article text. We ended up with the following heuristics:

- Covert all text to lowercase
- Collapsed spaces
- Strip punctuation (: ; “ - *) that’s not useful and split remaining punctuation marks from surrounding text. We tried to follow the same pattern here as the training data provided to us in PA1.
- Removed text blocks which contained any of the following words:
  - “copyright”
  - “current browser”
  - “all rights reserved”
  - “privacy statement”
  - “paid for by”
  - “terms of service”
  - “we are not liable”
  - “we experienced an error”
- Replace urls with a “[url]” token. The intent here is that we’re reducing sparseness because urls are not likely to be repeated but still attempting to preserve sentence structure and whether particular sources contain embedded citations.
- Remove single sentences (while treating ellipsis as a period) as well as sentences which contain fewer than 10 words. These heuristic were very useful for stripping out navigational text and comments embedded in the pages.
- Remove short text segments. A text segment was the unit of text defined by one <p> subtree. Note that for some forms of NLP analysis, this heuristic (as well as the one above) would bias the language of the news sources and have the potential to modify the perceived writing style (i.e. one news source may use lots of short sentences) but since we’re focused on n-gram language models, we don’t believe this bias towards larger blocks of text impacts our language models’ behavior significantly.

Our scraping and post-processing resulted in the following corpora for evaluation:

Source	Raw Data	Parsed and Processed Data
www.latimes.com	612MB	2.4MB
www.cnn.com	475MB	12.9MB
www.huffingtonpost.com	711MB	6.8MB
www.foxnews.com	126MB	1.7MB
www.nytimes.com	210MB	2.7MB
www.bbc.co.uk	165MB	3.2MB
www.msnbc.msn.com	545MB	14.8MB
www.washingtonpost.com	230MB	2.3MB
www.4biden.com	11MB	3MB
www.barackobama.com	20MB	1.6MB
www.ronpaulforcongress.com	5MB	152KB
www.johnmccain.com	5MB	137KB
www.jerrybrown.org	14MB	914KB
kucinich.house.gov	102MB	11.9MB
chrisdodd.com	5MB	85KB
freestrongamerica.com (Mitt Romney)	62MB	3.2MB
hillary4president.org	35MB	504KB

Future Work for Data Extraction Component

Even with our heuristics which aggressively strip short text (comments, tweets, etc), we observed that there was still some advertiser text in some of the extracted corpora. A future project could use the HTML markup to mitigate this effect.

Another possible area for future work would be to restrict the news sources just to their politics sections. Due to the naivety of our crawler, and the unstructured url-space of some news-sources, it was difficult to extract only politically-motivated articles from some sources. This made a direct comparison of the news agency content with the politicians content challenging given that some news agency sites had extensive content about topics not extremely relevant for politics such as entertainment and media.

We considered investing much more time here by extracting the relevant English text by hand for some of the pages and training a max ent classifier to label relevant English text in the rest of the raw corpora for us. This would be an interesting area for further work.

Analyzing Language Text

Our goal is to evaluate and compare the extracted corpora for different news agencies and presidential candidates campaigns using language models, with the intent of clustering the corpora according to nomenclature and parlance and correlating that with high-level characteristics such as political orientation.

We originally planned on building a classifier of text corpora into one of the two major political parties. However, we decided against building into our methodology this prior assumption of categories that comes with classification. Instead, we set off to use clustering techniques to try to observe clustering based on comparison of language models and correlate that with higher level properties of the different corpora. One particular area of investigation we had was whether certain news agencies corpora consistently cluster with a particular group of candidates with the goal of discovering which news agencies may follow a particular party, carry their message and/or target a similar audience with their publications.

Our approach to analyzing this problem involved building a language model for each corpus and then measuring the perplexity of each corpus in terms of each trained language model for each other corpus. The language model we used for this task is one we created for PA1 and dubbed the ZipfChimeraInterpolatedTriGramModel. This model utilizes EM to train a linear interpolation of a smoothed unigram model as well as bigram and trigram language models which each use Katz-Backoff. All of the models employ simple Good-Turing smoothing of the joint probability distribution by fitting a Zipf ar^b curve to the frequency of frequencies distribution. When evaluating enron and europarl corpuses as part of PA1 this model was the best performing model of those we evaluated.

Clustering

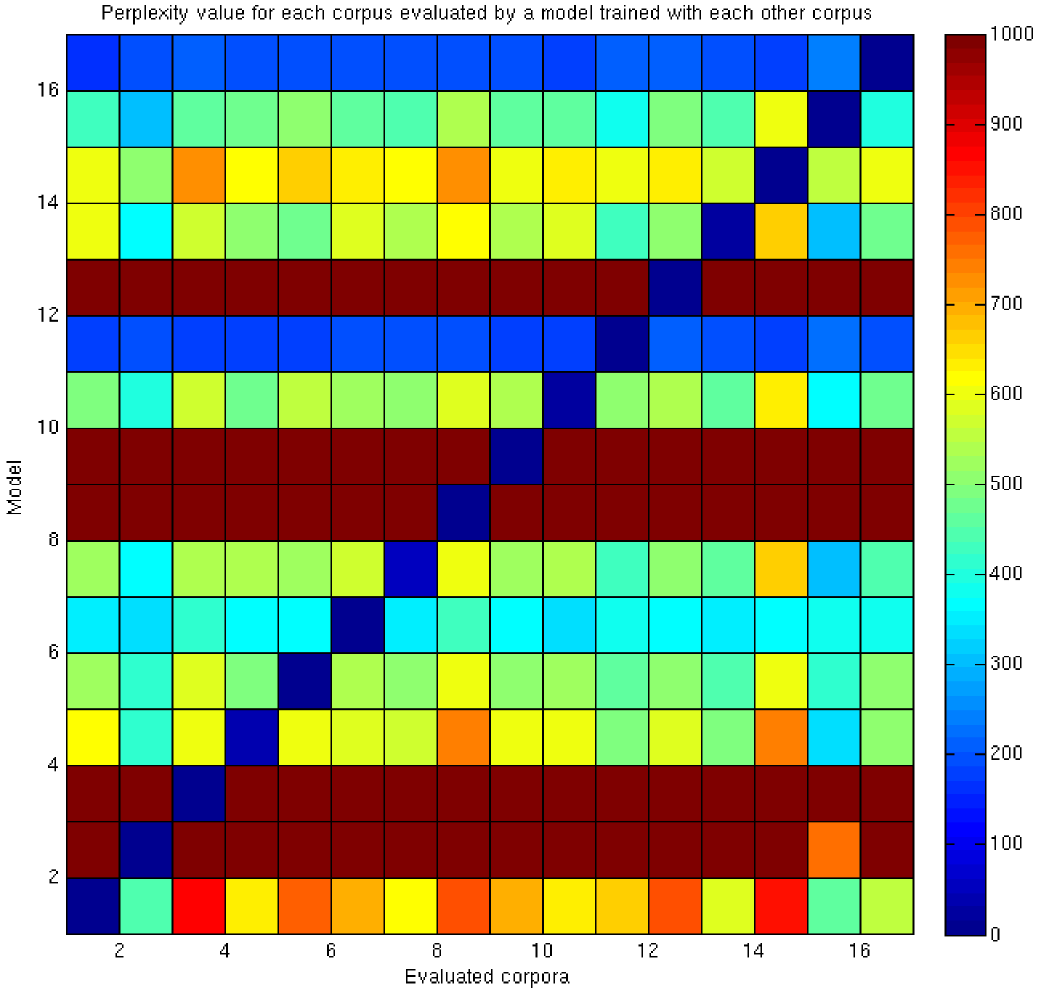
After obtaining the n-by-n matrix consisting of the perplexity of each corpus according to the language model of each other corpus, we applied a clustergram to several different subsets of that matrix.

Specifically, the clustering algorithm that we used consisted of first standardizing each row, meaning taking the perplexity values for each language model and subtracting the mean and normalizing so that the standard deviation is 1. It then applies a hierarchical clustering linkage algorithm that at each step joins the two closest clusters, using the average Euclidian distance between pairs of points in each cluster as the distance metric for any two clusters. This clustering method is applied to both the rows and columns of the matrix, and we visualize them by graphing the resulting matrix of columns and rows permuted to match the order of the leafs in a tree representation of the hierarchical clustering. Fortunately, most of this is functionality readily available in MATLAB through functions such as linkage [5] and dendogram [6].

Results

Figure 1 is a heat-map depicting the perplexity of each corpus evaluated relative to a trained language model for each other corpus. Each row corresponds to a trained language model for a corpus, and the columns correspond to the evaluated corpora.

Figure 1



Sources Legend

1. bbc	6. hillary	11. mccain	16. ronpaul
2. biden	7. huffingtonpost	12. mittromney	17. washingtonpost
3. chrisdodd	8. jerrybrown	13. msnbc	

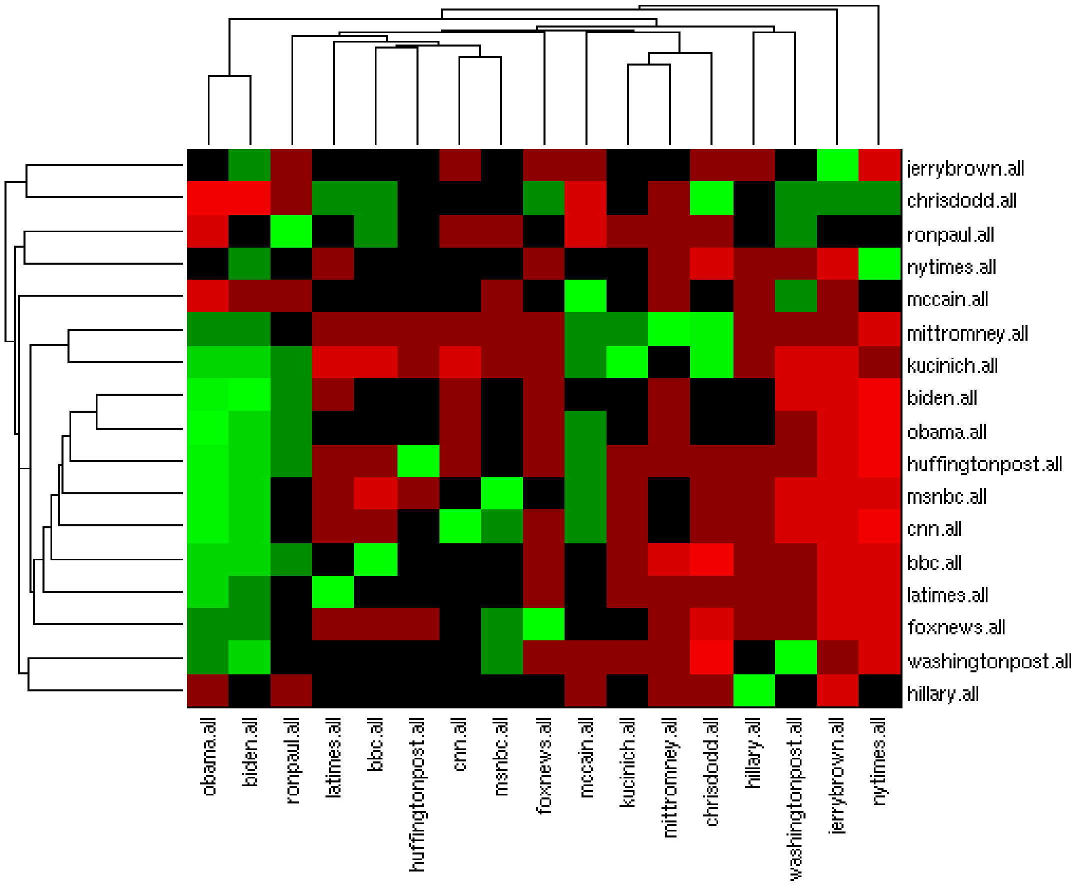
4. cnn	9. kucinich	14. nytimes	
5. foxnews	10. latimes	15. obama	

There are some interesting observations already apparent:

- While this visualization provides the most resolution for the majority of candidates it lacks some details and doesn’t provide much information about the biden, chrisdodd, jerrybrown, kucinich and mittromney corpora which all had perplexity above 1000 for the majority of the evaluated corpora.
- The only evaluated corpus which had a perplexity of less than 1000 for the biden model is the obama corpus. Similarly, biden’s corpus has the lowest evaluated perplexity for the obama model.

Figure 2 illustrates a clustergram of all the models and evaluated corpuses:

Figure 2


Clustering of all corpora evaluated by the models for each other corpus.

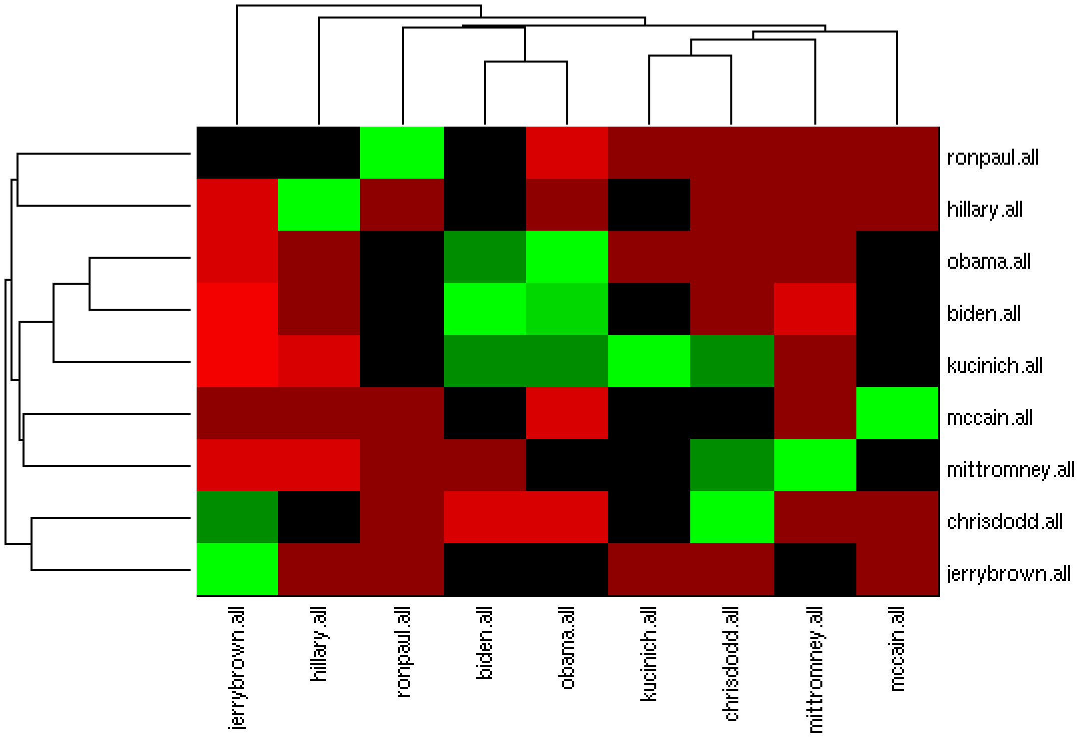
We begin to see some clusters forming. We observe that the strongest cluster is comprised of the obama and biden corpuses. We infer that there is some commonality in the text used by these two politicians who were part of the same ticket, and who focus on similar issues. It’s also noteworthy that while they cluster together, the raw perplexities of each corpus differ significantly (they differ by ~1000 on average as seen in Figure 1) so the actual measured perplexity varies considerably, but the relative perplexity across the evaluated corpora is similar.

We also learn that most of the news sources are the next most-likely corpora to cluster with the biden and obama corpora. This stands to reason from them being the president and vice president, and their actions and platform are likely to be newsworthy. Here we see that the NYTimes does not cluster as well with the rest of the news sources. Since we were not able to download the entirety of the NYTimes and analyze its content, it’s possible that we did not obtain as much political content as some of the other news sources.

We narrow our focus solely to presidential candidates in an attempt to find relevant clusters yielding the next clustergram:

Figure 3

Clustering of candidates corpora evaluated by models for each other candidate corpus.

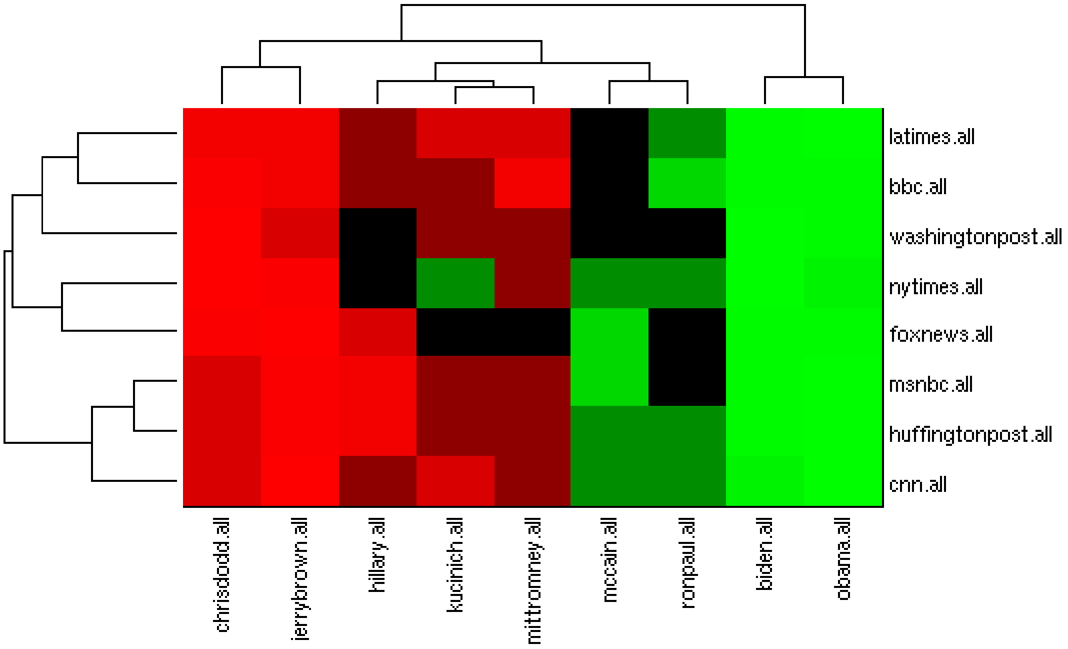


As before, we see a tight clustering of obama and biden. We also observe that while not all the candidates cluster into two groups along party lines, we do observe clusters of candidates that correspond with political parties. In particular, looking at the clustering of the language models (the rows) we see obama, biden and kucinich, another democrat, form one cluster. Another cluster exists between chrisdodd and jerrybrown, other democrats, and another cluster exists between mccain and mittromney. The aberration here is the clustering of the ronpaul and hillary corpora. We took a cursory glance at the hillary corpus to see if we could find anything abnormal. To our dismay, we clicked through some articles and quickly realized that the website we crawled to generate the hillary corpus was actually a content farm. The first few links we clicked were articles that appeared to be politically motivated so we assumed they were from Hillary’s platform and we didn’t explore the site too carefully. As such, we didn’t look further into this apparent clustering.

Our next focus was to examine the relationship between the news agency corpora and the candidate’s corpora. Figure 4 depicts the news corpora as evaluated by the candidates’ corpora, whlie Figure 5 depicts the inverse:

Figure 4

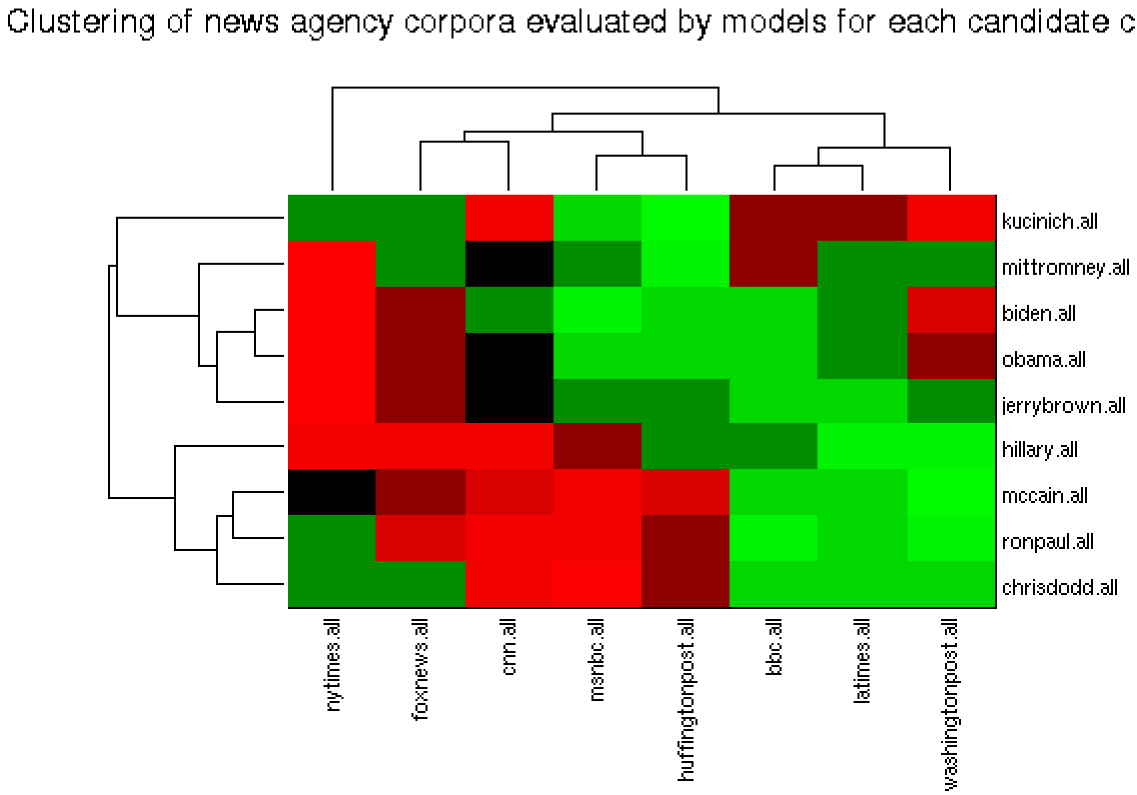
Clustering of candidates corpora evaluated by models for each news agency corpus.



From figure 4 we observe that there isn’t much difference in the relative interpretation of each candidate’s corpus by the models trained from the news agency corpora, or put another way: there doesn’t appear to be a strong bias for either party by any news source. Interestingly, the language model representing Foxnews, which many claim to have a pro-republican bias [7], did not have a lower perplexity correlated with republican candidates. It’s worth noting however that in Figure 5, depicted below, the interpretation of cnn and foxnews by the candidate’s language models is clustered.

All of the news sources found obama and biden’s content to be very unperplexing, but they also didn’t find the corpora of mccain and ronpaul perplexing. Perhaps the difference between the clusters is the same as described above, where the actions and viewpoints of President Obama and Vice President Biden are commonly newsworthy. Also of note: all of these national and international news sources had higher perplexity when evaluating the text of the corpora for senators chrisdodd and jerrybrown.

Figure 5



Finally, in Figure 5 we see two notable clusters amongst the language models. There's a cluster for mittromney, biden, obama, jerrybrown (and to a lesser-extent kucinich), all of which represent Democrats aside from the mittromney corpus. There's a second cluster with hillary, mccain, ronpaul and chrisdodd, which ignoring the hillary corpus for reasons stated previously, in the majority case represents Republican candidates. There also appears to be a correlation between the candidates hillary, mccain, ronpaul, chrisdodd and the news agencies bbc, latimes and washington post. We don't have a strong intuition for this observation. We attempted to understand these finding (and that of cnn and foxnews clustering together) by analyzing the distinguishing features of the corpora as described in the next section.

Understanding the Clustering - Word Clustering Analysis

We found some interesting clusters in our analysis above, but we wanted to have a better intuition about the reason why certain corpora were clustering together. Ultimately, all the data used above boils down to comparing the 1,2,3-gram occurrences in each pair of corpora, so we elected to take a deeper dive in comparing the language models themselves, and we started by specifically looking at the unigram distributions.

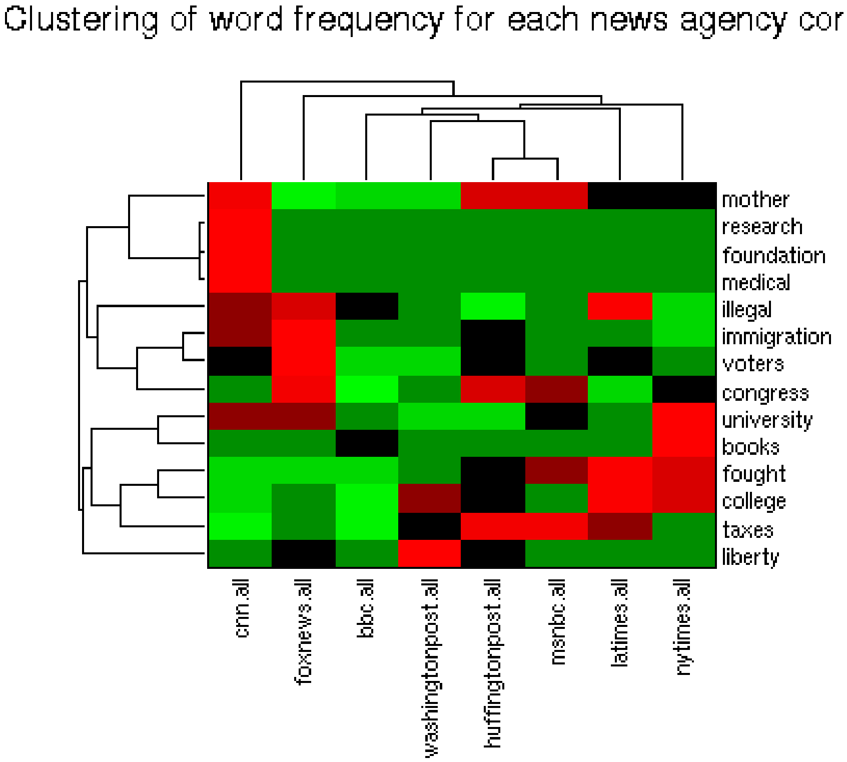
On a first naive approach, comparing unigram distributions is hard because the head of the distributions tend to be very similar, containing the same most common English words, and the tail being full of single-occurrence words that don't provide much insight. We decide to look specifically for the words whose occurrence varies the most across all corpora, which allowed us to ignore the standard common English words without having to move too far away from the head of the distribution.

More specifically, we wrote a script (find_important_words.py) which computed the unigram distributions for each corpus, throwing away words with too low frequency (< 0.0001), and then for each of the remaining words, computing the min and max frequency for that unigram in each of the corpora. We then sorted all the unigrams in decreasing order of the ratio of these max and min frequencies across all the corpora and considered the top 50 ones.

The set of these 50 words by itself was very interesting, many of them being words that express a political connotation. Of these top 50 words, we manually inspected them and selected 14 that we judged most interesting from the point of view of depicting political connotation. They are: mother, research, foundation, medical, illegal, immigration, voters, congress, university, books, fought, college, taxes, liberty.

Figure 6 and 7 depict the result of clustering the set of news agency corpora and candidates corpora according to the frequency of these 14 words in each of the corpora. Note that we chose to have words in the rows due to how the standardization of the clustering works. It made intuitive sense to standardize for each word.

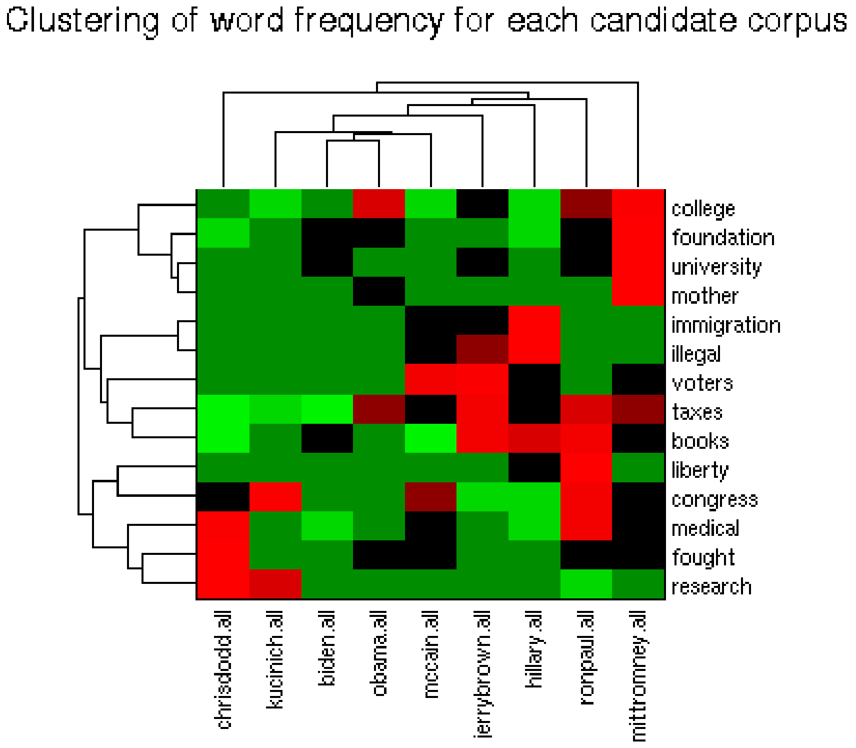
Figure 6



A couple of features in the clustergram in Figure 6 were noteworthy. We observed that the cluster of words (university, books, fought, college, taxes, liberty) were responsible for dividing the set of news sources into two sets, one in which these words were less common (cnn, foxnews, bbc) and the other set where at least some subset of these words were more frequent. Interestingly, the clustering of cnn and foxnews in figure 5 where we evaluate the news sources according to the language models based on the candidates, gives us a hint that this set of words, a subset of the language models for the candidates corpora, plays a significant role in why cnn and foxnews cluster together in figure 5.

Another interesting comparison between the clustering of news corpora based on candidate corpora language models (figure 5) and select words frequency (figure 6) is that huffingtonpost and msnbc were a tight cluster in both, and in the case of word frequencies, the top 3 words that showed a particular high frequency for these two corpora compared to the rest of the news agencies were “taxes”, “congress” and “mother”.

Figure 7



Clustering of word frequency for each candidate corpus

Finally, in figure 7, once again the obama and biden corpora represent the tightest cluster. However, there was no particular high occurrence of any of the select words in both of these corpora, and the fact that none of the select words had a high frequency, except “college” for biden, was likely the reason why they clustered together. Going back to figures 2 and 4 where we observed that biden and obama were observed to have low perplexity, particularly among the news agency corpora, seems to be compatible with this observation - the corpora for obama and biden seem to be significantly closer to an “average” language model across all corpora we observed and thus why we don’t see many of the select words with a particularly high frequency. Allowing ourselves some extrapolation, we think this is a reflection of the fact that many of the issues and topics covered in the obama and biden corpora are just more aligned with general issues news agencies cover.

Most notably, we see that the top cluster of words (college, foundation, university, mother, immigration, illegal, voters, taxes and books) defines two significantly different sets of candidates, one with chrisdodd, kucinich, biden and obama, 4 out of the 6 democrats, and one with mccain, jerrybrown, hillary, ronpaul and mittromney, 3 republicans and 2 democrats, one of them being hillary which has some of the data source issues mentioned previously. That is a significant separation of candidates according to party affiliation and it gives us some intuition about the clustering of candidates based on party affiliation that we observed in the analysis of figure 5 and others.

Conclusions

We observed evidence of correlation of the low-level statistical 1,2,3-gram based language models' data with high-level corpora features like political affiliation. In particular, we saw a consistent grouping of the obama and biden corpora as well as some significant clustering of candidates of same-party affiliation. However, we did not see two strong and distinct clusters for the two political parties in any of the data we analyzed. Moreover, we did not find evidence of political bias from any of the news agencies with respect to either party, as the clustering of evaluated perplexity of candidates was fairly uniform across all news source language models. Regarding the news agencies, we did not see any strong feature characterizing the clusters of news agencies we observed in the data.

Future Work

Clustering of the news sources and candidates' platform to look for common parlance and nomenclature is really just the tip of the iceberg in this area. Given more time and resources, identifying relevant political issues and performing sentiment analysis to address bias or positioning towards these issues would be an interesting area for research.

References

[1] Heritix - http://crawler.archive.org/
[2] WebSPHINX - http://www.cs.cmu.edu/~rcm/websphinx/
[3] wget - http://linux.die.net/man/1/wget
[4] BeautifulSoup - http://www.crummy.com/software/BeautifulSoup/
[5] - Create agglomerative hierarchical cluster tree http://www.mathworks.com/help/toolbox/stats/linkage.html
[6] Dendogram plot - http://www.mathworks.com/help/toolbox/stats/dendrogram.html
[7] http://www.fair.org/index.php?page=1067

Statement of Collaboration

We collaborated together on all aspects of the project. The code was all pair-programmed and the report was generated together.