“A Nation Divided” : Classifying Presidential Speeches

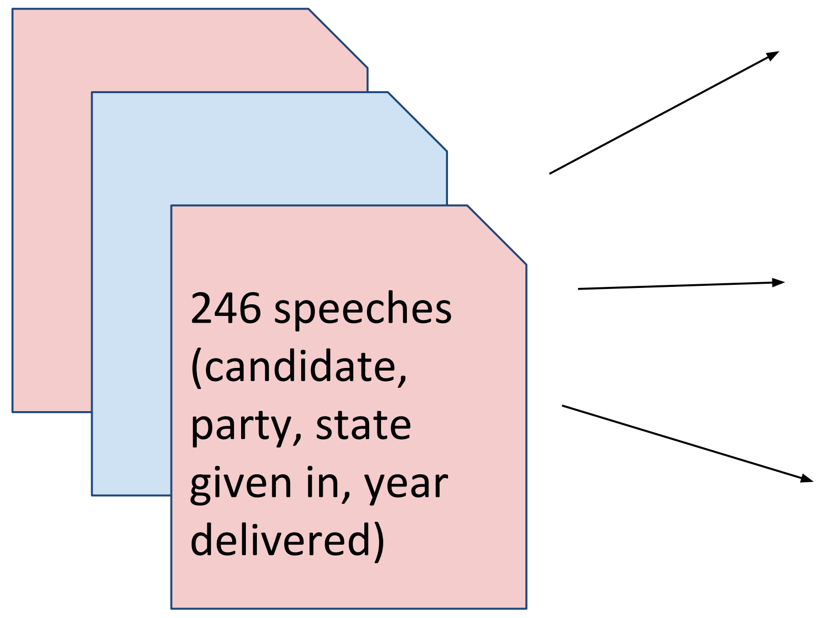
Ambika Acharya, Nicole Crawford, Michael Maduabum
{aacharya, nicolecr, rmaduabu}@stanford.edu

MOTIVATION

Politicians are often accused of changing sides on issues or going along with the party’s stance. We wanted to see if politicians’ speech differed greatly depending on their party affiliation, or when or where they were speaking. Using machine learning models, we were able to successfully predict the party of the speaker and the election year in which a speech was given based solely on the text. We also tried to build a model to predict the region of the country a speech was delivered in and had moderate success.

PREDICTING

Given a presidential candidate’s speech, can we predict their party affiliation, the location of the speech, and the year the speech was given? We use the predictive algorithms SVM and Logistic Regression and defined our tasks as followed:



246 speeches (candidate, party, state given in, year delivered)

Party Prediction:
{Democrat, Republican}

Region Prediction:
{South, Coasts, Midwest}

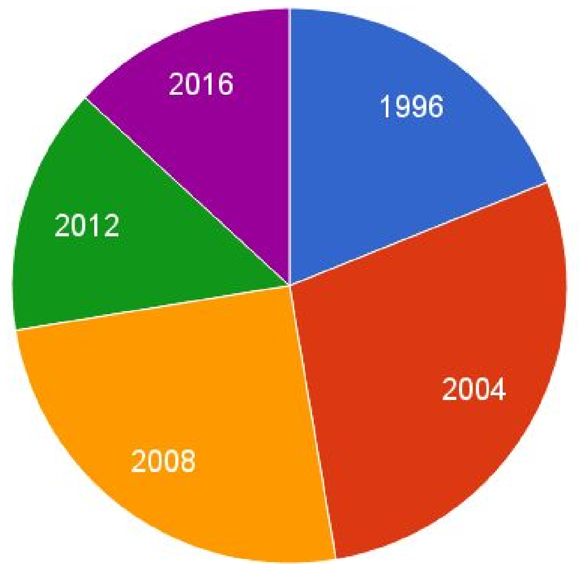
Election-Year Prediction:
{1996, 2004, 2008, 2012, 2016}

We also used Naive Bayes to do **topic clustering** to see which words are often associated with speeches given by Democrats vs. Republicans, amongst specific regions and over time.

DATA SET

Our dataset contains full speeches by Presidential candidates speeches from The American Presidency Project.

- 1996-2016 presidential elections
- Democratic & Republican non
- 246 total speeches
  - 129 Democrat
  - 117 Republican



METHODS

Classification Models

Since predicting party(Dem/Rep) is a binary classification task, we decided to compare the performance of logistic regression and SVM models:

- Logistic Regression

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

- Support Vector Machines

$$\varphi_{\text{hinge}}(z) = [1 - z]_+ = \max\{1 - z, 0\}$$

For the multiclass tasks, predicting election year and predicting which region of the country a speech was given in, we used the One vs. Rest versions of these models. This method trains a binary classifier for each potential class that distinguishes it from all the other labels. Additionally, we used L2 regularization and 10-fold cross validation to prevent overfitting.

Feature Engineering

For all tasks we used unigrams and bigrams: both counts and tf-idf weighting. We removed stop-words and stemmed words. We did feature engineering for each task to improve model performance:
Party: speech length
Region: election year, party, is swing state, keywords by region
Election-Year: same as above and keywords based on current events

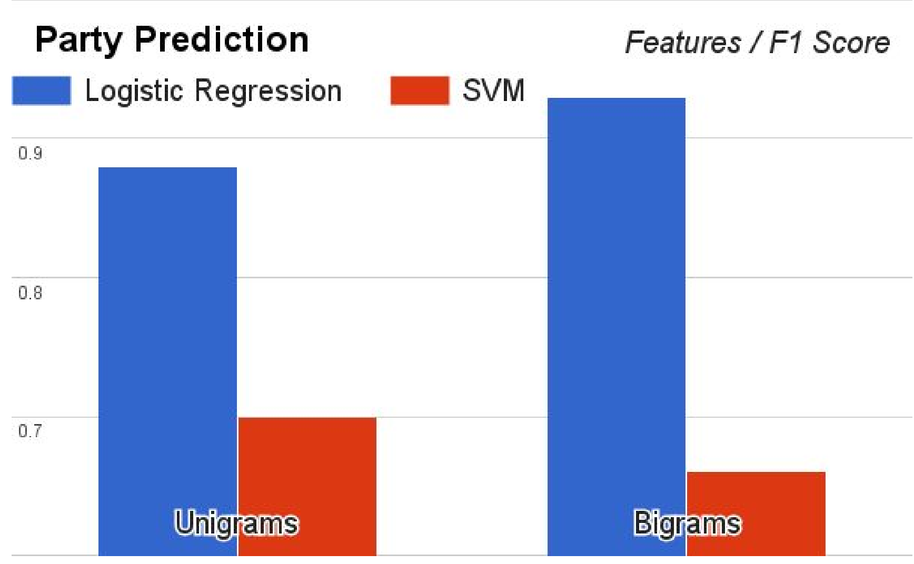
Topic Clustering

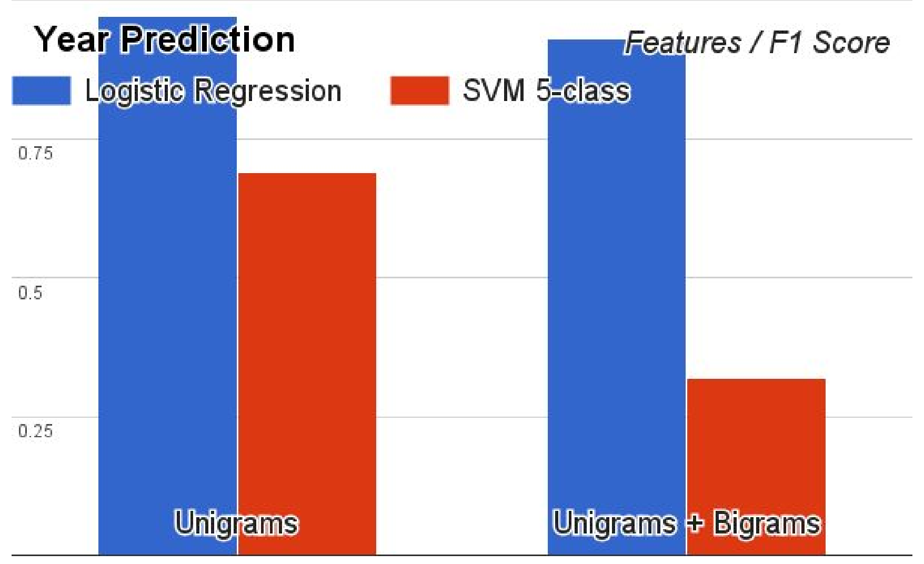
We used a Naive Bayes model to model class probabilities and use those to find words highly associated with a given party, region and time period. We used the following expression for class probability.

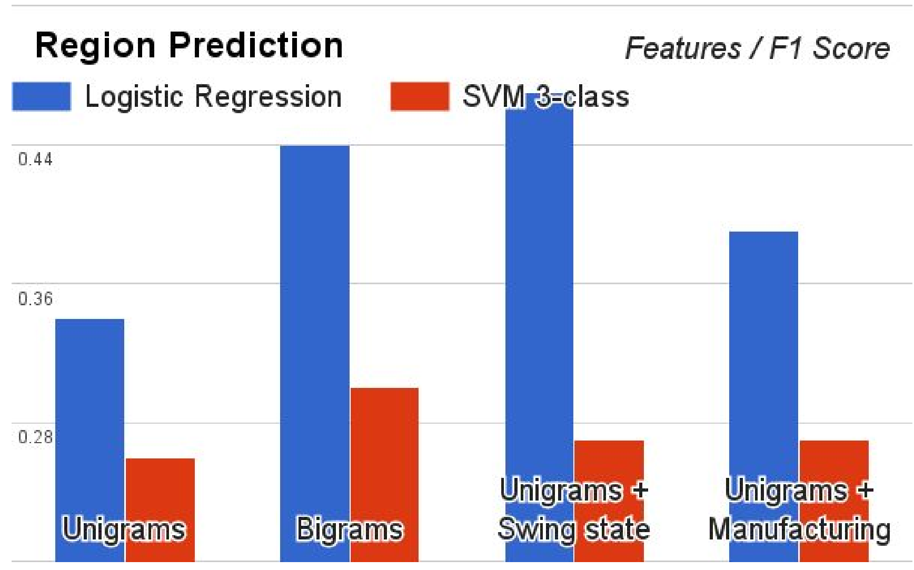
$$p(y = 1|x) \quad = \quad \frac{\prod_{i=1}^{n} p(x_i|y = 1)p(y = 1)}{\prod_{i=1}^{n} p(x_i|y = 1)p(y = 1) + \prod_{i=1}^{n} p(x_i|y = 0)p(y = 0)}$$

RESULTS

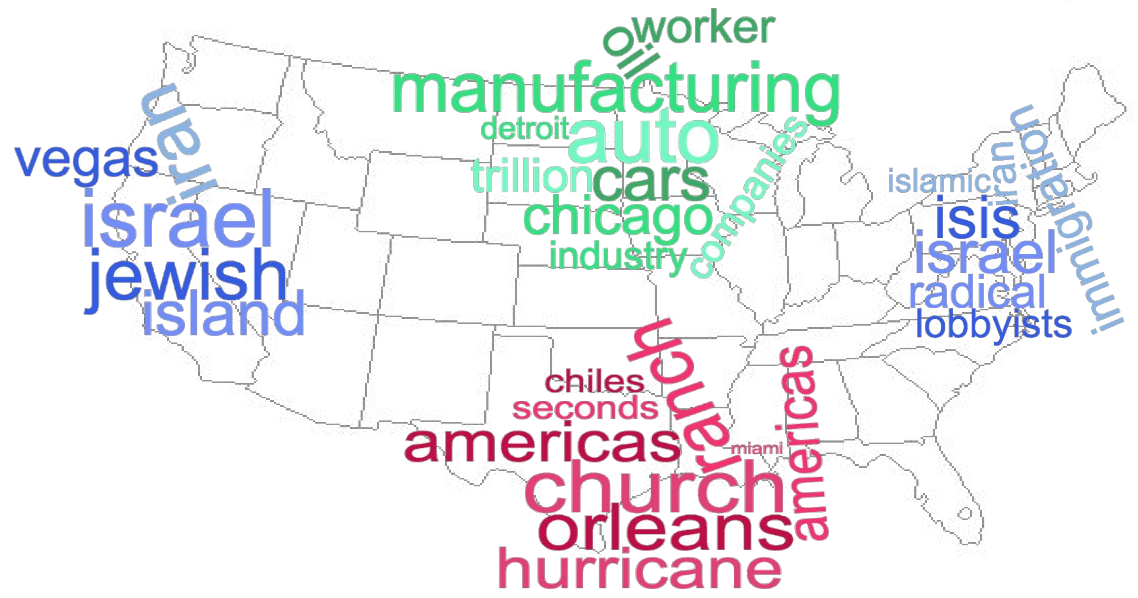
We report results of F1 scores using 10-fold cross validation:







Words most predictive in speeches per region



ANALYSIS

Our results were very good (~95% accuracy) for the party prediction task which demonstrates how distinct the rhetoric of the two parties is. The topics discussed in speeches seem to differ significantly between republicans and democrats which allows for easy unigram/bigram based separation. The year prediction task was also very successful. We manage to get over 90% accuracy choosing between 5 classes for unseen speeches. Again this is because the topics discussed in different election years varies widely. Predicting the region of the country a speech was given in proved much more difficult which implies that similar ideas are discussed across different regions.

FUTURE WORK

Next steps include using word2vec and sentiment analysis for our models and using ablative analysis to find the most influential features. Additionally, we would try and experiment with different models such as neural networks to improve classification.

REFERENCES

Natural Language Processing with Python (Bird)
Lexical cohesion analysis of political speech (Klebanov)
Classifying party affiliation from political speech (Yu)
Thank you to the CS229 Professors & Course Staff!
Wayne Chang