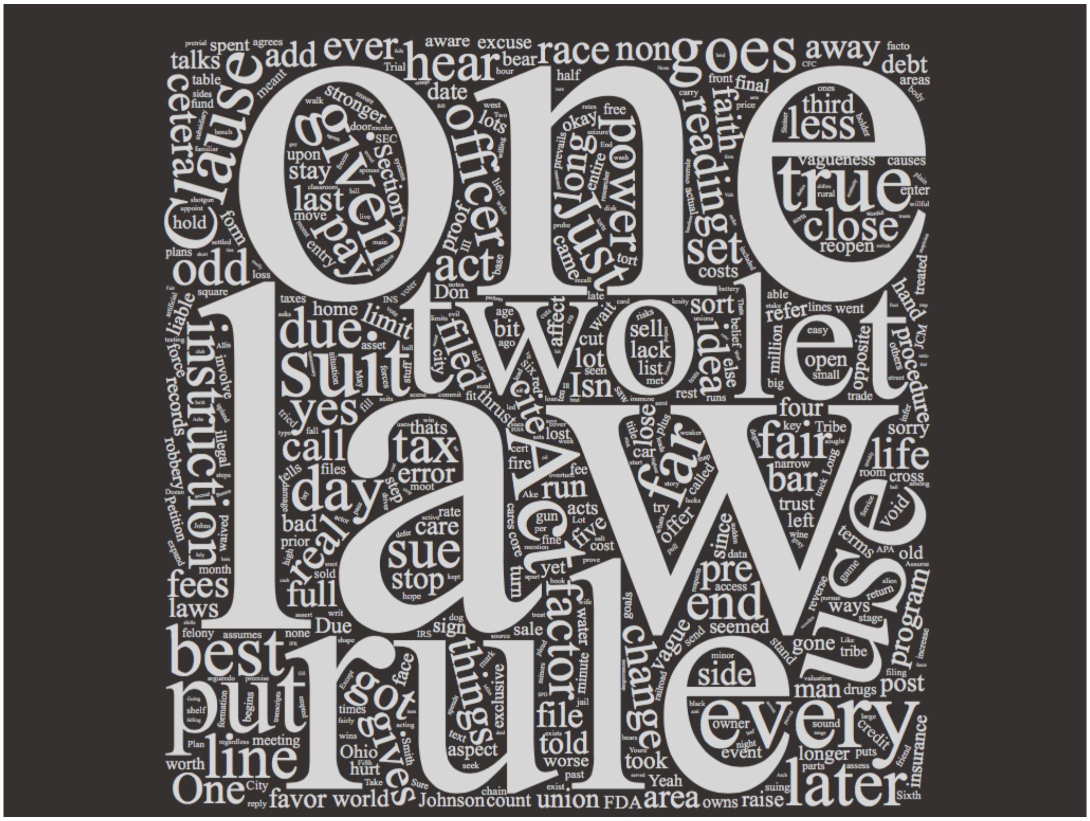
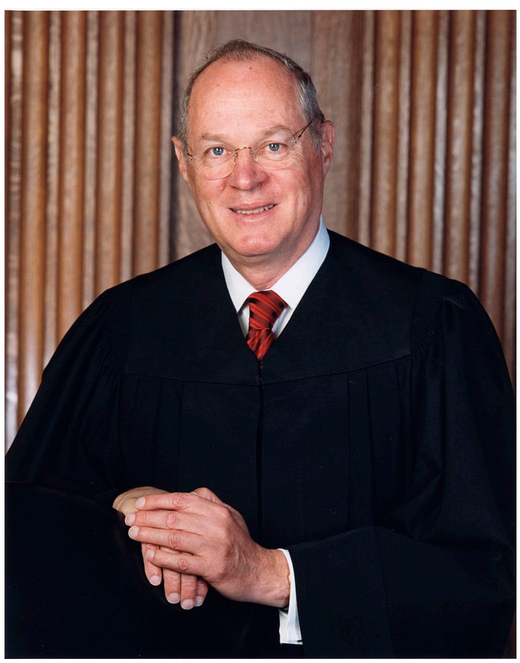
The Voting Patterns of Supreme Court Justice Anthony Kennedy





Introduction

Justice Anthony Kennedy has been considered a "swing vote" on the Supreme Court of the United States (SCOTUS) since Sandra Day O'Connor retired from the Bench in 2005 and is currently the subject of much interest in the legal sphere. Since the SCOTUS is comprised of 9 individuals, Justice Kennedy can provide the key vote in determining contentious cases when the other 8 Justices are deadlocked 4-to-4[1].

I was inspired to learn more about Justice Kennedy's voting habits after reading a FiveThirtyEight article[2] that details how a Ph.D. student named Chris Nasrallah attempted to model Justices' speech in order to find any patterns in their voting. While he was relatively successful at building out correlative features that pertained to how specific Justices voted, he only achieved an overall accuracy of ~70% against a baseline of ~68%; this means he only had a predictive power of roughly 2%.

While Nasrallah focused on how the SCOTUS together would vote (all 9 Justices), I was more interested in Justice Anthony Kennedy's voting patterns. Would it be possible to predict his votes using speech text alone? Would other features ultimate be more important? And is it easier to predict his voting patterns if we isolate and examine specific genres of law rather than examining it monolithically as Nasrallah did?

Practical Applications

Although this analysis is not strictly applicable to a business problem, it has the potential to provide valuable insight as to how Justice Kennedy (and by extension, the other Justices) will vote in the cases that come before them. Practically speaking, this would be useful to a small handful of individuals, corporations, and lawyers who have business before the court. For example, lawyers who argue cases before the Court could tailor their oral arguments to the particular tendencies of each Justice or alter their oral argument mid-presentation if they sense that they are not ingratiating their argument with any particular Justice.

More generally, the results of these findings may be of interest to public advocacy groups which have business before the Court that can have an enormous impact on American society. For example, *Obergefell v. Hodges* was the landmark 2015 case in which Justice Kennedy was the key swing vote in legalizing same-sex marriage. Examining key voting patterns and the way Justices speak to lawyers prosecuting their case before the Court may offer valuable insights on how to better frame their arguments.

---

[1] http://www.pbs.org/weta/washingtonweek/web-video/justice-anthony-kennedy-decider
[2] https://fivethirtyeight.com/features/how-to-read-the-mind-of-a-supreme-court-justice/
[2] https://fivethirtyeight.com/features/how-to-read-the-mind-of-a-supreme-court-justice/

Data Wrangling

Gathering the Data

Perhaps the most difficult portion of this project was capturing the text data in order to make predictions and evaluations. All of the transcripts necessary to complete this project were published in PDF format on the Supreme Court's website. In order to capture the text data, I laid out a plan to first scrape the PDF documents for the website, parse the PDF documents to extract the text, and finally cleaning the text data for exploratory data analysis and modeling.

In order to build my PDF scraper for the Supreme Court's website, I had to learn how to use the following python packages: os, Regular Expressions, UrlLib, Requests, Beautiful Soup. Once I had a strong grasp of each package and the Supreme Court's website, I built a webscraper that grabbed each PDF transcript of oral argument available, dating back from 2001 all the way up unto 2017. Each PDF file was named using the Docket Number (format: YY-Case No.) for later processing in the text mining process.

After downloading and organizing all of the PDF transcripts into folders by year, I mined all of the PDFs for their text content; to do this, I learned how to use the PDFMiner python package and wrote a script to execute to complete the job. Unfortunately at this step, I recognized two distinct problems with the files I had downloaded.

The first problem was that the PDF transcripts for years 2000-2003 were password encrypted and could not be converted from PDF format to text. After fruitlessly trying numerous times to locate the password or a way around the encryption, I decided to discard these transcripts.

The second problem I encountered was that PDFMiner did not correctly translate the text for the transcripts in years 2013 and 2014; the returned text was strung together without any spacing between words. After searching for a solution that would add the correct spacing back into the text files without any luck, I also decided to pass on these particular years and discarded them from the dataset.

This left me with a set of text files with the transcripts for Supreme Court Oral Arguments between the years 2004-2012 and 2015-2016. From here, I wrote another script that filtered out miscellaneous text and corporate markings from the court reporting company that prepared and published the oral argument transcripts.

The University of Washington in St. Louis SCOTUS Database

In order to parse the data that I gathered, I need to find an external database to pair Kennedy's speech with case details. Luckily, I found the University of Washington in St. Louis SCOTUS Database[3], which provided the critical information that I needed to conduct my analysis. The database tracks every single Supreme Court ruling since its inception and has numerous labels and features for each case.

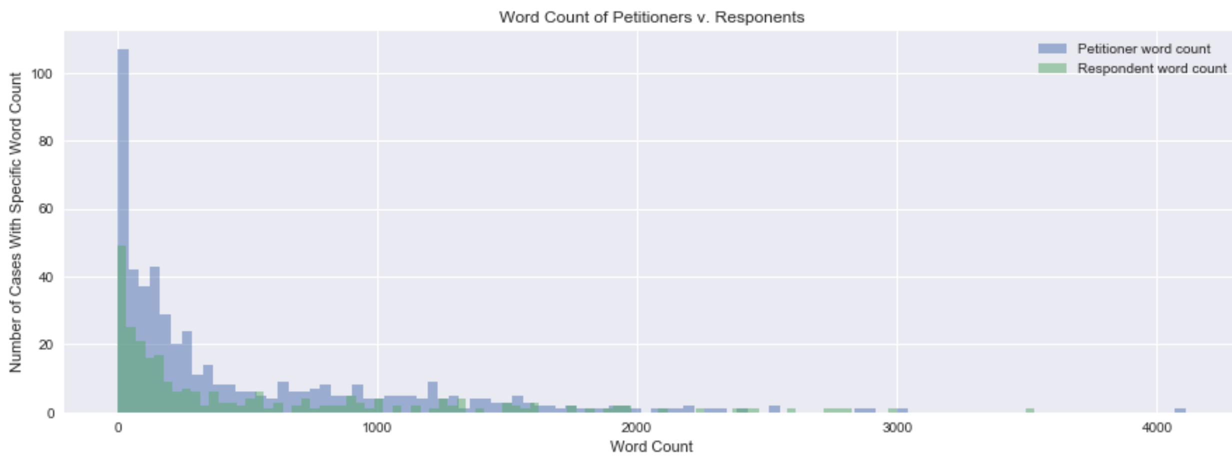
---

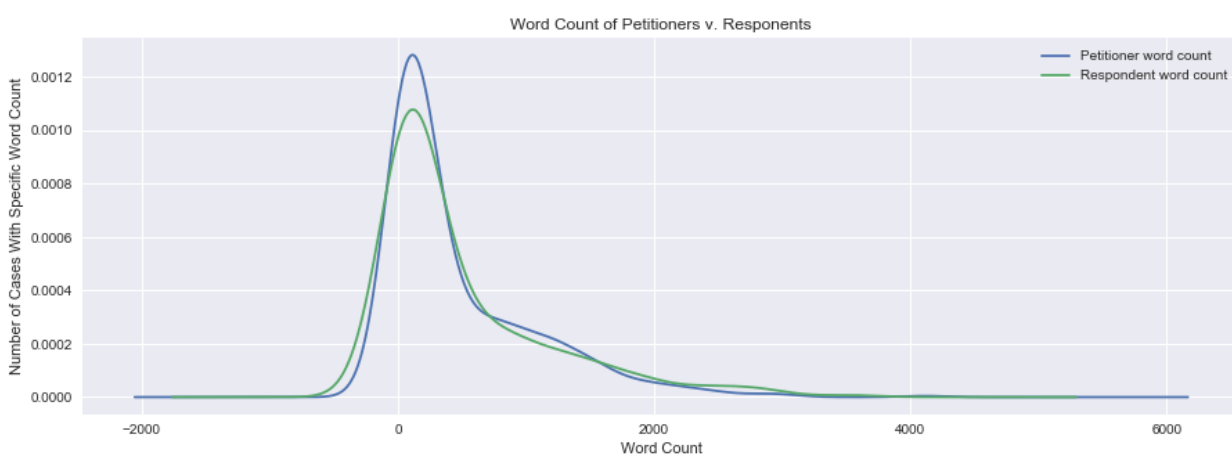
[3] http://scdb.wustl.edu/

Although Justice Kennedy has been on the SCOTUS since 1988, I could not take full advantage of the WUSTL database due to the fact that useful/accurate transcripts for the Court's oral arguments only went as far back as 2004. With more data, I could potentially run further analysis on the progression of Justice Kennedy's full career on the Court.

Exploratory Data Findings

The first item I examined was word count and its relationship to how Justice Kennedy voted. Court observers often make hay over how Kennedy's oral argument speech may affect his vote.[4] To try and see if there was any correlation between his voting patterns and word count, I performed a T-test. Not surprisingly, word count was only weakly correlated with how he voted. However, it should be noted that Kennedy more often stays silent (meaning no speech during oral argument) when he votes for the Petitioner than the Respondent.

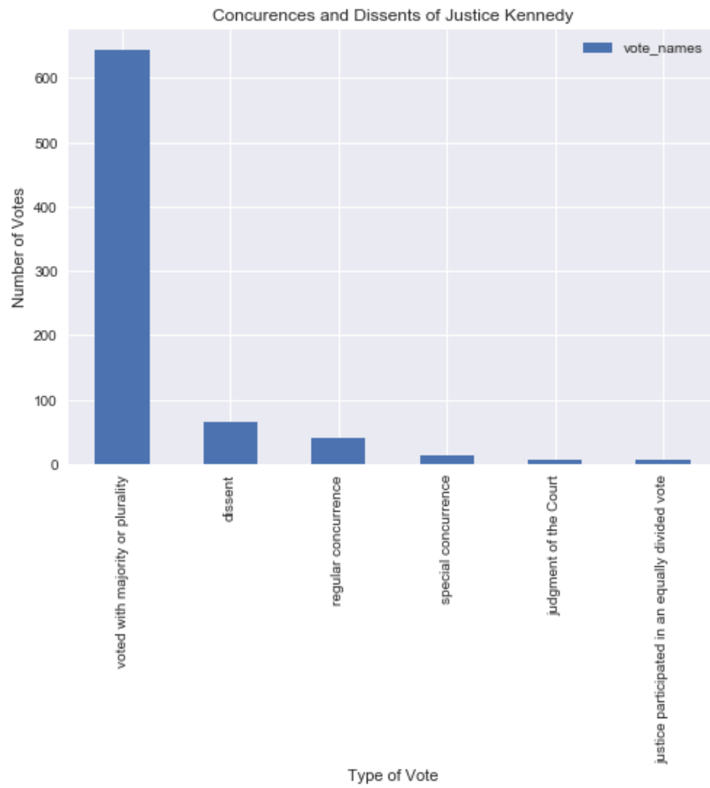


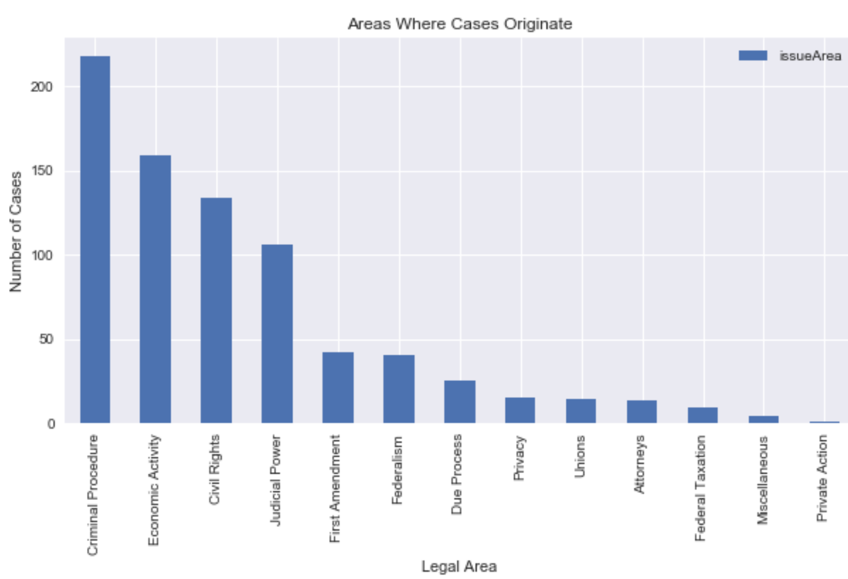


---

[4] https://fivethirtyeight.com/features/what-justice-kennedys-silence-means-for-the-future-of-gerrymandering/

Other areas that I examined included Kennedy's vote type, meaning did he vote with the majority or with the minority and specifically in what ways. The data shows that Kennedy rarely dissents and usually votes with the majority/plurality or writes a concurrence with the majority/plurality. Furthermore, I also looked at the different types of cases that come before the SCOTUS: Most deal with criminal procedure followed by economic activity, civil rights, and judicial power.





Machine Learning Results

The main objective of my Machine Learning model was to try and build a predictive engine to determine which side would win the case based on text speech. The feature that was fed into each model was text speech that had been stemmed using the Snowball Stemmer and then run through a TF-IDF vectorizor; the label was which side won (the petitioner/plaintiff or the respondent/defendant).

I decided to break down the cases by Issue Area before running each set of Machine Learning Models, since running the entire dataset against each model made little sense. Law is broken into distinct areas and different Justices may have habits or particular quirks on one area that might allow predictions in some areas over others. I chose to review Issue Areas that had at least 40 cases in them. The results show that this approach was the correct one: Different areas of the law had significantly more (or less) predictive power than others, a fact that would have otherwise been obscured.

The top-performing model was the Civil Rights issue area analyzed by the Multinomial Naïve Bayes. Over the baseline score (simply the number of cases where random guessing would provide the same predictive power), it had a positive predictive power of 7.9%. The lowest performing model was the First Amendment issue area analyzed by the SVM model. Unfortunately, it had a negative predictive power of 16.6%; random guessing using the fraction of winners and losers would have been a better choice in this instance.

Predictive Power Based on TF-IDF Vectorizations of Stemmed Speech Text

	Baseline	Multi. Naive Bayes	SVM	AdaBoost	Decision Tree	Random Forest
Criminal Procedure	65.1%	68.8%	64.2%	51.4%	52.2%	66.9%
Civil Rights	73.6%	81.5%	76.9%	76.9%	70.8%	78.5%
Economic Activity	70.1%	77.2%	77.2%	59.5%	58.2%	72.2%
Federalism	56.4%	45.0%	45.0%	50.0%	50.0%	45.0%
First Amendment	69.0%	66.7%	52.4%	66.7%	66.7%	66.7%
Judicial Power	58.5%	52.8%	52.8%	52.8%	52.8%	54.7%

Conclusion

The type of case with the highest potential for predicting the winning side is civil rights. Multinomial Naïve Bayes paired with a stop-word removal tool, the Snowball Stemmer, and Tf-Idf yield a decent model with which to predict outcomes. However, the majority of other areas of law yielded little to no predictive power when compared with Baseline measurements. This indicates that generally speak, speech text alone (albeit, text that has been cleared of stop-words and stemmed) does not have much predictive power. Other features such as word count, number of interruptions, and/or average length of word could potentially offer more predictive power when combined with speech text.

Although my predictive power using speech text alone was not as high as I had hoped (I was hoping for +10% predictive power), my best model was higher than that of my inspiration for my work, Chris Nasrallah, who only was able to obtain a predictive power of +2%. Overall, a predictive power of +8% for a specific issue area is acceptable and in hindsight, expected. Perhaps Mr. Nasrallah could improve the predictive power of his model by focusing it first on specific Issue Areas and optimizing from there.

Areas of Future Work

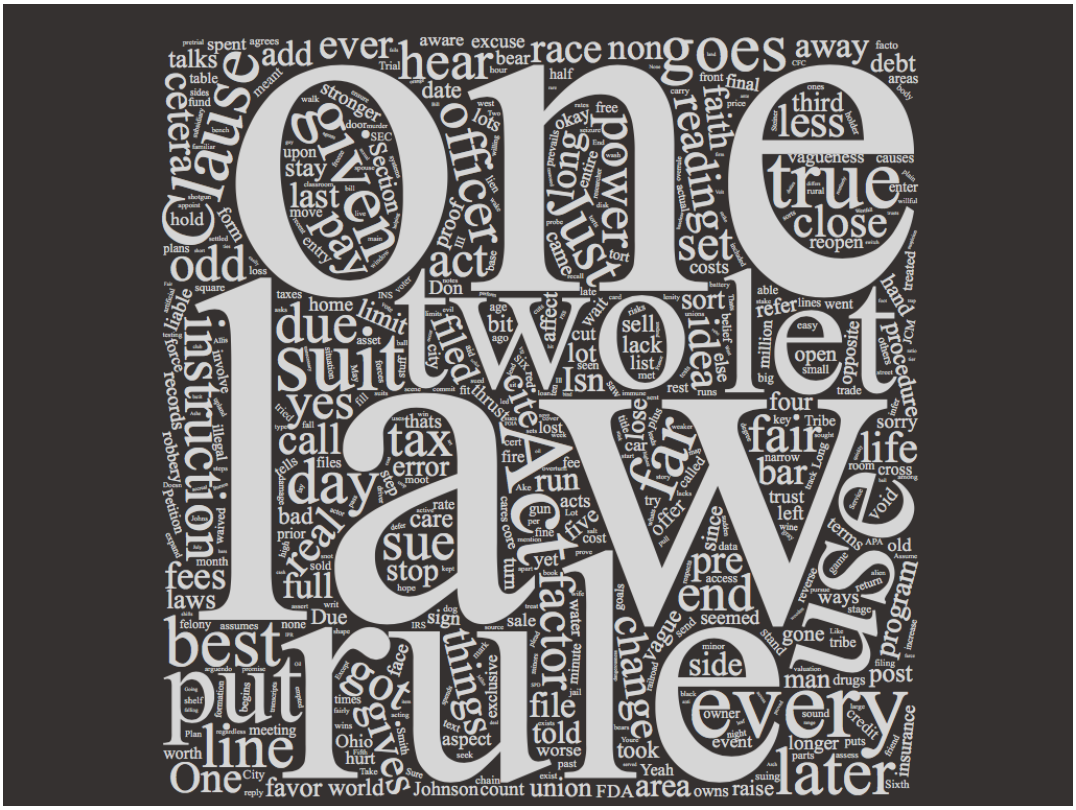
A few hurdles that I encountered that could potentially be overcome with more time and expertise:

1. Cases between the years 2000 and 2003 were dropped because they do not differentiate between Justices while they are conducting oral arguments. The transcripts simply label speech as "Justice", which for my purposes is unhelpful. The only way to potentially correct this problem would be to use recordings of the oral arguments in the Oyez Project to label the speech. This would be time prohibitive and I would only undertake such a project if it could be automated in some way.
2. Cases in years 2013 and 2014 were dropped because the PDF to text file conversion left all of the words joined together without spacing. After searching for a solution to break the strings/text apart, I was only able to come up with a partial solution that could have polluted the speech data. Instead of risking pollution, I decided to eliminate this data. Given more time, I may be able to research and/or engineer a solution to this particular problem.
3. Engineering more features. Unfortunately I was only able to engineer word count into my dataset. In the future, with more time, I would like to engineer more features such as the number of times Justice Kennedy interrupted the plaintiff or the petitioner, the number of times Justice Kennedy is interrupted by the plaintiff or the petitioner and the number of times he speaks over other Justices and vice versa.
4. Joining multiple features to text speech to gain higher predictive power. For example, I was unable to get word count or number of interruptions paired effectively with speech Justice Kennedy spoke.
5. Try to build a separate model without oral argument speech as its primary basis and merge it with oral argument models in an ensemble method to boost predictive power. An example of such a separate model can be found here[5]

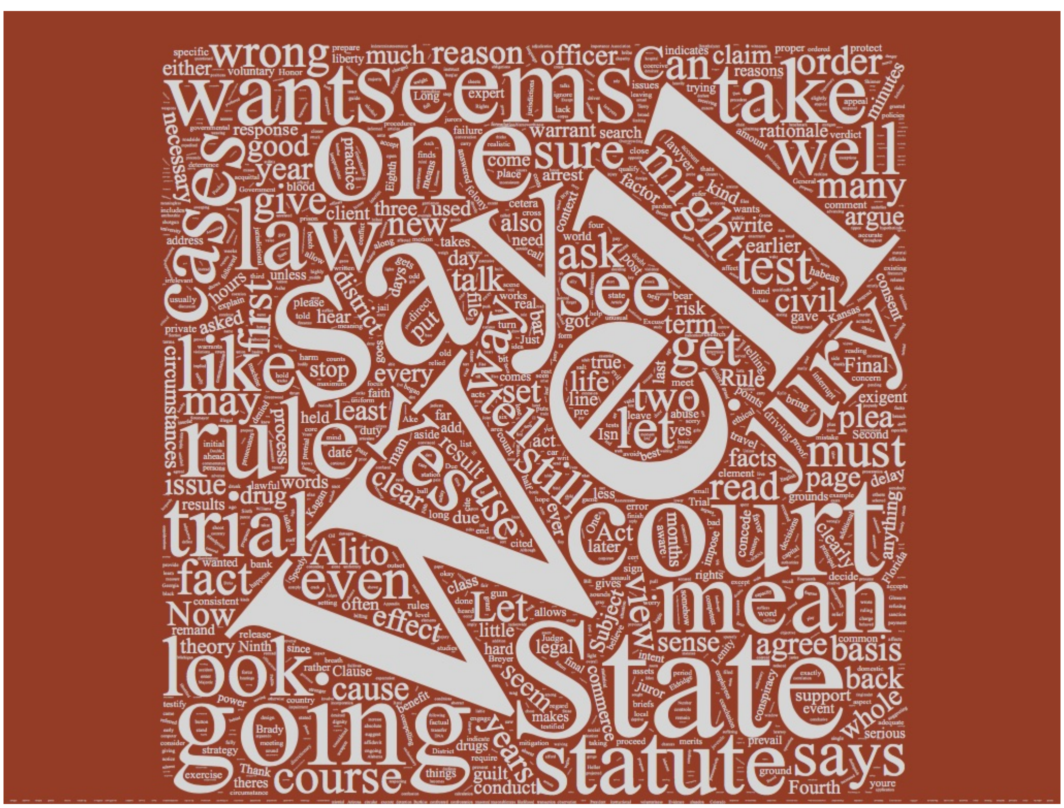
[5] https://www.vox.com/2014/8/4/5967147/how-a-computer-model-got-to-predict-70-of-supreme-court-decisions

Word Clouds from the Six Issue Areas That I Examined

Word Cloud: All Aggregated Speech



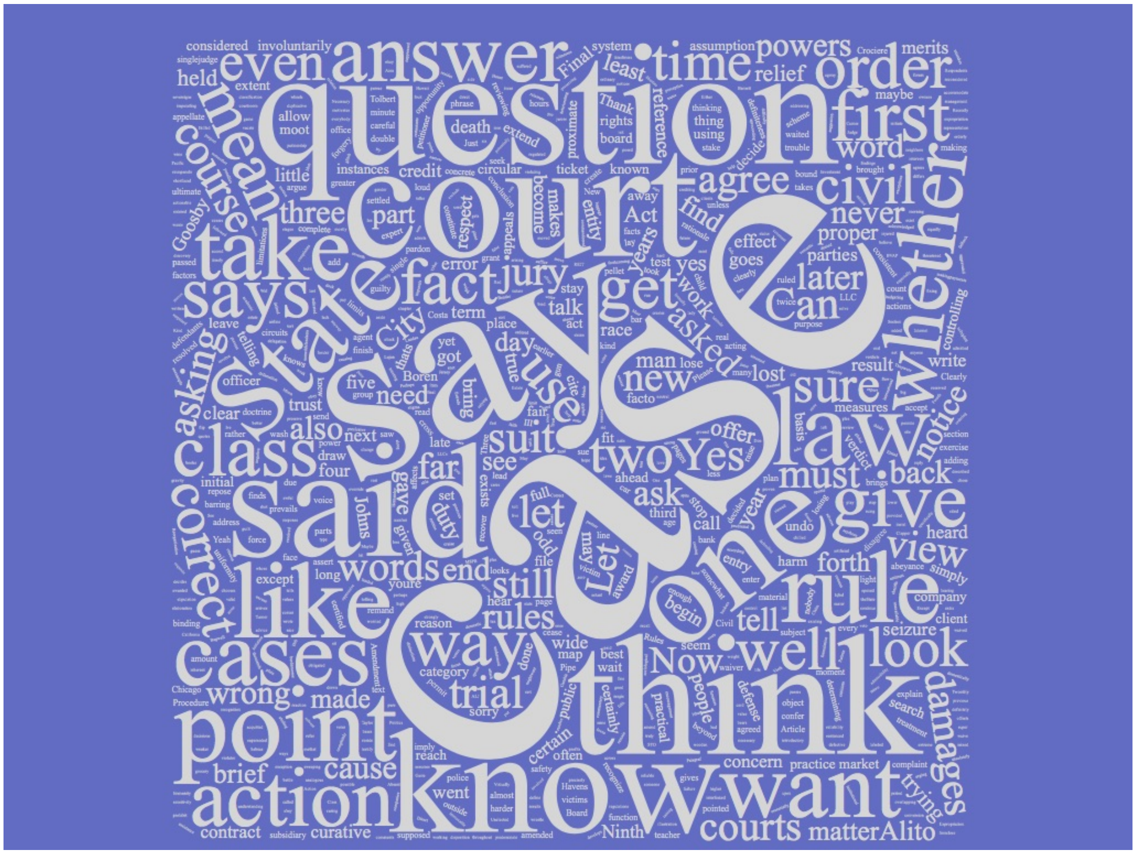
Word Cloud: Criminal Procedure



Word Cloud: Economic Activity

Word Cloud: Civil Rights

Word Cloud: Judicial Power



Word Cloud: First Amendment

Word Cloud: Federalism