Using NLP to Predict the Behavior of the Supreme Court of the United States

W266 Final Report - Fall 2017

Alex Jamar, Omar Al Taher, Peter Zhou

Abstract

Our work focuses on applying natural language models to Supreme Court case transcripts. The research we present is predicated upon the assumption that there are discernible patterns within the language used in the oral arguments presented at The Court. We experiment with a number of different machine learning techniques to predict case outcomes based on these patterns. Our work investigates the relative merit of bag-of-words and bag-of-vectors representations of the case proceedings, which we use to train Random Forest, Naive Bayes, and Logistic Regression models.

Introduction

Predicting the behavior of the Supreme Court can have many implications. Political scientists and legal experts have long presented and debated their predictions. The decisions of the court have a major impact on American society. Therefore, the ability to predict those decisions in advance could have a number of uses, depending on the case. Our project will attempt to predict the verdicts of the U.S. Supreme Court based on the oral arguments of the cases. Since court proceedings involve complex arguments, many parties with different speaking styles, and are based on a very large legal code, this topic represents a significant challenge.

Background

The current most referenced work on this topic was done by Katz et al., (2014) which achieved a 70.2% accuracy for the case outcomes and a 71.9% accuracy for the justice votes. The work used random forest classifiers on case “metadata” such as the lower court decisions, justice, term, court of origin, etc. This work did not analyze the actual proceedings or other textual data. Sim et al., (2015) use a natural language processing (NLP) approach to predict the Supreme Court decisions based on the amicus briefs filed for each case. A prediction accuracy of 74.3% is reported but the results are harder to interpret here, as the prediction was for “the most likely partitioning of the justices into two groups” as opposed to a direct prediction of the verdict.

Other studies have used NLP to analyze court proceedings, for example on the French Supreme Court (Sulea et al., 2017) and the European Court of Human Rights (Aletras et al., 2016). Even though these two studies analyze court systems that are quite a bit different from the US Supreme Court, they are considered a form of a baseline as they have the similar approach of using the textual features of court records to predict the court’s decision. Sulea et al use SVMs trained on bag of words and bigrams representations of the cases to achieve remarkable results with a maximum accuracy of 96.9%. It is noted in that paper that the French judicial system is known to be more predictable than the United States judicial system which

may partially explain this result. Aletras et al has a very similar approach of using SVM trained on a bag of n-grams representation of cases, and achieve an accuracy as high as 0.76. Due to demonstrated results of this approach on problems from two different legal systems using different languages (French and English), we consider this approach as a reasonable starting baseline for our project.

One difference of note is that the representations of the cases in both papers rely on the final published results of each case, as that was the only available data available to them. Different approaches are used to "mask" the results from the classifier, but this is inherently different than the case representations we use (the oral arguments) which reflect events occurring prior to the court's decision.

Data and ETL

We began our project by scraping the oral argument transcripts from the SCOTUS website. After several attempts to transform the data into a clean and usable format, we decided to look elsewhere for the transcripts. We found the Oyez Project, which contains an archive of all audio recorded in the court since 1955. Along with it, there is an API that allows access to a transcript file and a summary for each individual case in json format. Files for a total of 7,899 cases were processed and a loaded into a single dataframe with each row representing a "speaking turn". Each "turn" is labelled with case title, speaker name, speaker role (justice, petitioner, respondent), appointing president (if justice), start & stop time, and the text. The final dataframe has a total of just over 1.5 million "turns" and is made available as a collection of csv files.

We also used data from the Supreme Court Database from the Washington University School of Law that contained files with additional information such as the "winning party". The database codes a "Winning Party" for each case as follows:

- The petitioning party lost, coded as 0 (court affirmed lower court decision or dismissed the case/denied the petition)
- The petitioning party won, coded as 1 (court reversed, reversed and remanded, affirmed and reversed in part, affirmed and reversed in part and remanded, or vacated)
- Result unclear, coded as 2 (court certified to/from a lower court)

After accounting for blank json files, missing advocate data, and advocates where the 'lawyer side' was not available, we were left with 2,770 usable cases. For each case, the text was grouped into three groups, representing the texts spoken by petitioners, respondents, and justices (as a group). A simple join was performed for each speaker group, creating a single string per case/speaker combination.

Results and Discussion

The petitioner has received a favorable verdict 63.18% of the time. We felt this was an appropriate baseline for us to test whether our models would have any predictive power. We

began by using a count vectorizer bag of words representation with a logistic regression classifier that yielded results around 56%. It made sense to us that this would lead to a weak result because the texts contained many repeated words that did not add any meaning to the case. For example, in almost all cases each side's advocate starts with the phrase "Mr. Chief Justice," and "may it please the court." This and other court formalities mean that a method is needed to have the model ignore phrases and words that occur in many cases.

We eventually found that a logistic regression classifier applied to TF-IDF trigram model with features like number of turns the speaker spoke and the speaking length offered us the best result. Somewhat surprisingly, eliminating stop words using the built in functionality in scikit learn's TFIDF Vectorizer degraded performance. A simple stop words list is used consisting of 9 tokens. This model gave us 65.81% accuracy calculated by k-folds cross validation. The TF-IDF trigram model allows us to see if there are important phrases used by the lawyers and justices in different outcome scenarios. What we found out, as we will discuss later in the error analysis, is that the trigrams used in each outcome are not significantly different. One possible explanation for this is that the vocabularies in the two different categories of cases (the petitioning party losing vs. winning) are not materially different. In that case, no bag-of-words model would be able to improve upon a model that simply predicted the most common outcome for every test case.

This led us to attempt using the Gensim implementation of the Doc2Vec model that would theoretically be able to encode more information than a bag of words representation. This model was trained on the complete dataset of 1.5 million "turns". This includes data excluded from our classifier model due to the lack of speaker labels. The training data is also included in this step, based on the fact that this is an unsupervised learning model and the data is not labelled in any way. The Dic2Vec model was using the distributed memory algorithm with a 5 word window. Different embedding dimensions were tried at 100, 300, 500, and 1000. Slight improvements in word similarity were apparent as the embedding dimension was increased, with a tradeoff of large memory requirement for loading the model (greater than 6GB for the 1000 dimension version) and longer compute time at inference. An embedding of size 500 was used as a good compromise.

Evaluation of the word similarities generated by the Doc2Vec model was encouraging. Some examples of the most similar words according to the embeddings are shown below:

Input Word	Doc2Vec most similar words
running	walking, runs, flying, moving, ran, run
lawyer	lawyers, attorneys, counsel, client, attorney, cellmate
Scalia	souter, ginsburg, kagan, sotomayor, breyer, stevens
beautiful	strap, antagonisms, bothersome, gabaret, boggled, dirty
sad	enured, flasher, hulbert, catchy, silvermaster, sour

The model does well on word level embeddings (especially in the legal domain), matching lawyer to lawyers, attorneys and counsel. It also detects that Scalia, Souter, Ginsburg and Kagan are all similar. However, some results point to an interesting interpretation of words like beautiful and sad. Analysis of the performance of the model on document level embeddings was more challenging and outside the scope of this paper.

After training the Doc2Vec model, we removed stop words and punctuations from the case data before embedding and training a logistic regression classifier on it. This gave us an accuracy of 65.09% which is slightly worse than the bag of words TF-IDF model. This result was counter-intuitive based on the intuition that embeddings can capture similarities between words where a bag of words model cannot.

Error Analysis

As indicated earlier, our research was predicated upon the assumption that patterns in oral arguments presented before the court could be used to predict case outcomes. In reviewing our results, we conclude that neither the Bag of Words model nor the Bag of Vectors model successfully picked up on the nuances of the language used in each category of case.

With regards to our trigram model, we analyzed the top 100 trigrams present (defined as those that occurred at least once per case) in six different vocabularies: the petitioner's, the respondent's, and the supreme court justices' when the petitioner won, and likewise when the respondent won. For both the petitioner and the respondent, there was nearly a 90% overlap between the top 100 trigrams when the petitioner won vs. when the respondent won. In other words, we found virtually no difference in the most common speech patterns when one party won vs. when the other party won.

However, we did surface a few interesting findings after we eliminated the most common trigrams present in either both the petitioner vocabularies or both the respondent vocabularies. For example, the trigram "no further questions" appeared either at the very top or near the top of the vocabularies for the winning party, but was not present among the most common trigrams of the vocabularies for the losing party. We interpret this to mean that the lawyers in those cases felt confident in their position and therefore used the phrase "no further questions" to emphasize that they had made their point successfully. Though that particular finding (and others)

appeared to be more than mere chance, evidently they did not have sufficient statistical importance to drastically improve the performance of our models.

Future Work

The main challenge we ran into was in the ETL portion. We ended up being able to only use 2,770 out of 7,899 cases due to missing, corrupted, or mislabeled data. If we had more time, we could increase our dataset by up to 65%. We could also potentially improve the accuracy by performing sentiment analysis or clustering to generate additional features. Another approach would be to combine our work using NLP with Katz's feature engineering research to hopefully generate an even better model.

We also feel that an LSTM could provide great results if trained on the turn-by-turn data since it would be able to take advantage of the meaning present in the sequencing of the words spoken. The very nature of oral arguments means that almost every statement is a response to a previous one. A lawyer's previous arguments don't vanish when they make the next one. An LSTM could provide a framework that could make connections between the words of different speakers or phrases used in previous arguments from the same speaker. We feel that this fact may potentially boost predictive power.

Even though our models did not output drastically better results than the baseline, we were still able to show that natural language processing does have some power predicting the outcome of Supreme Court cases.

Datasets:

1) Supreme Court of the United States Oral Transcripts, 2000-current day.
   Two hundred pieces of information about each case decided by the Court.
   Format: pdf document
   https://www.supremecourt.gov/oral_arguments/argument_transcript/2017
2) The Supreme Court Database, "Metadata" 1946-2016. Washington University Law
   Two datasets available
   a) Case-centered data: case level information: each row corresponds to a dispute
   Format: csv file
   b) Justice Centered Data: justice votes for each dispute, plus case information
   Format: csv file
   http://scdb.wustl.edu/data.php
3) The following set of annotated Supreme Court Transcripts retrieved by github user Walker Boyle (username walkerdb) on May 30, 2017 from the oyez.org public api. Data is made available using the Creative Commons Attribution-NonCommercial 4.0 International License https://www.oyez.org/license
   Format: json
   https://github.com/walkerdb/supreme_court_transcripts

Reference Papers:

1) Daniel Martin Katz, Michael J. Bommarito II, and Josh Blackman. 2014. Predicting the Behavior of the Supreme Court of the United States: A General Approach. http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0174698
2) Yanchuan Sim, Bryan R. Routledge, Noah A. Smith. 2015. The Utility of Text: The Case of Amicus Briefs and the Supreme Court. http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/download/9361/9558
3) Martin AD, Quinn, KM, Ruger, TW, Kim PT. Competing approaches to predicting supreme court decision making. Perspectives on Politics, 2004 http://www.wusct.wustl.edu/media/man1.pdf
4) Octavia-Maria Sulea , Marcos Zampieri , Mihaela Vela , and Josef van Genabith. 2017. Predicting the Law Area and Decisions of French Supreme Court Cases https://arxiv.org/pdf/1708.01681.pdf
5) Nikolaos Aletras, Dimitrios Tsarapatsanis, Daniel Preotiuc-Pietro, and Vasileios Lampos. 2016. Predicting Judicial Decisions of the European Court of Human Rights: A Natural Language Processing Perspective https://peerj.com/articles/cs-93.pdf