Forecasting Justice: An Examination of Supreme Court Oral Arguments' Predictive Power

Jonas Heim, Matt Kaufmann, Michael Wagner, Núria Adell Raventós, Sergio Olalla Ubierna

Github repository link

Abstract

The Supreme Court plays a crucial role in shaping public policy and influencing societal norms through its judgments on critical issues. Thus, there is a strong interest in predicting the outcome of the cases before the ultimate ruling is made. By leveraging pre-trained models such as BERT, RoBERTa, BigBird, and LongFormer, we attempt to predict Supreme Court case outcomes and judges' voting patterns.

The study utilizes relevant datasets obtained through ConvoKit, extracting key features and eliminating unnecessary metadata. Our transformer models, specifically RoBERTa and BigBird, performed the best. RoBERTa, despite having a token limit of 512, performed the best on predicting the outcome of cases at an accuracy rate of 68.57%, which is an improvement over the baseline of 3.04%. Furthermore, BigBird was able to outperform the baseline of 80% of the justices. Our CBoW model performed the worst, confirming our hypothesis that contextual information and semantics are important in predicting outcomes, compared to the choice of words in isolation.

Throughout the implementation of this study, we were limited by computational resources and thus there are several ways to improve the models we used. For instance, instead of using pre-trained embeddings we could look at training context-specific embeddings, improve data nuance by capturing more detailed information such as the interaction dynamics between speakers, and perform a comprehensive grid search to fine-tune the hyperparameters.

Introduction

The Supreme Court is one of the most important institutions in public policy. By scrutinizing and rendering judgments on critical issues, the Court influences the civil fabric of society. While legal scholars and practitioners have attempted to predict case outcomes and the votes of justices based on legal theories, the recent advent and advancements in machine learning provide us with an innovative lens through which to predict the outcomes of Supreme Court cases and the voting behaviors of judges.

Our project combines a number of techniques in the field of Natural Language Processing (NLP) to predict Supreme Court case outcomes and judges' votes. Leveraging data at the utterance level, we employ pre-trained models such as Bert, RoBERTa, BigBird, and LongFormer which have demonstrated superior performance in understanding the context and semantics of lengthy, complex texts – a characteristic that makes them especially suitable for legal texts.

To capture the language complexities inherent in legal language, we develop a bag of words neural network model. The Bag of Words (BoW) and continuous BoW models, while traditionally simple, have proven effective in many NLP tasks, particularly when embedded in a more complex neural architecture. By creating an embedding space for the words found in the Supreme Court utterances, our models are designed to learn and represent the intricate relationships between different legal terms and phrases.

Finally, we integrate sentiment analysis into our methodology, using a pre-trained sentiment library to determine if the sentiment expressed in court utterances has any correlation with the outcome of a case. This approach recognizes that beneath the seemingly objective nature of legal proceedings, emotions can play a key role in the court's decisions.

The project's core objective is twofold: (1) to predict case outcomes in favor of either the petitioner or the respondent, and (2) to predict the voting patterns of a subset of judges. Based on how well our models predict a given judges vote, we can potentially understand how much predictive power utterances have on judges' decisions, or the impact of oral arguments on the judges’ perspective of a case. These findings will not only contribute to legal studies but could also have far-reaching implications for how we understand the dynamics and decision-making processes within the Supreme Court.

Literature Review

We adopt the framework introduced by Medvedeva et al. so that we can characterize the space of prediction in court decisions into three different categories (Medvedeva et al. 2022). This research can be broken down into outcome identification, outcome-based judgment categorization, and outcome forecasting. Specifically, these methods can be confined to the classification of court decisions.

Outcome identification is the task of identifying the verdict within a case. In this case, the entire text related to the case can be used. The key differentiator in this prediction task is that explicit references to the verdict itself can be used as training data. We aren't trying to forecast the outcome, but we are instead attempting to classify the decision based on the full corpus of the case content. One use case for this method is to produce an automated classification system, as many courts around the world don't utilize structured information to account for the verdict. Various studies have been published confronting this classification task, using a myriad of machine-learning models. Some papers have reported accuracies approaching 99%.

Outcome-based judgment categorization is related to the prior task of outcome identification but with a key difference. Like outcome identification, outcome-based judgment categorization includes any information published within the final decision or judgment but excludes explicit references to the verdict in the judgment. In this task, since the verdict is already known, its principal usefulness comes from the identification of predictors within the case material. For this reason, algorithms used to attempt this task mustn't be a 'black box'.

Outcome Forecasting utilizes textual information about a given court case which was available only before a verdict was made public. Our models in this case are not 'seeing' any information about the final judgment of a case. Here we assume that no input information has been influenced by the outcome of the case. In contrast to the other two tasks, outcome forecasting allows us to make predictions about future court verdicts that have not been made as robustly yet.

In terms of classification, the research in outcome-based judgment categorization tends to perform worse than that of the outcome identification. This is intuitive because we remove explicit references to the court's decision. The authors (Medvedeva et al. 2022) look at 14 different outcome-based judgment categorization articles and conclude that only 3 of them successfully accomplish meaningful feature extraction.

When we look at the articles which attempt to forecast outcomes, we see significantly reduced classification accuracy compared to the prior two methods. For example, in the case of the articles using the supreme court dates, the accuracy level only reached 70%. This figure is only 2 percentage points better than a naïve prediction in which the petitioner always wins (Sharma et al. 2015, Katz et al. 2017). Outcome forecasting seems to be a very difficult forecasting task, in which modeling human behavior brings complex challenges.

Interestingly, in the paper by Katz et al. (2017), a random forest model was used to predict (i) affirmation or reversal of status quo judgment; and (ii) how individual justices voted. So although this paper does not use NLP, we can still learn from the challenges the authors ran into for their outcome forecasting, which is what we are hoping to do. Of note are the guiding principles the authors used to informed their model development, which will also inform our own:

- Generality: Factors such as public pressure, court composition, etc. both influence outcomes and change over time. So, building a model that is general enough to handle these factors is crucial to predictions.
- Consistency: We must factor in the bias-variance trade-off and not overfit on training data so that changes in external factors do not reduce model effectiveness.
- Out-of-sample applicability: We must only utilize data available prior to a decision date so as not to inform predictions with results that would not be available at the time of the prediction

Additionally, Katz et al. (2017) also ran into the interesting problem of figuring out the null hypothesis (baseline). There were a couple of options:

- Common wisdom among the legal community states that the baseline should be always guessed reverse. It is true that since 1940, the Supreme Court has had rulings reversed more often than approved, but the opposite was true before then.
- Another approach is to guess the most frequent outcome with an infinite or a finite memory. While the prior simply computes the frequency of outcomes up to a certain date since the first ever ruling of the Supreme Court, the latter uses a moving average. The issue with an infinite memory is that it would still predict affirm today so the authors generally conclude that a most frequent outcome with a memory of 10 years is optimal.

Given that we focus on the 20 years between 2000 and 2019 where reverse has been the most frequent outcome, both approaches described above would make predicting reverse as our baseline. Note that this was observed in other papers such as Hawes (2009) where they developed baselines for decisions as well.

Aside from general principles and baseline comparisons, Katz et al. (2017) had to think through some considerations around both outcome variables and feature engineering. For example, the outcome variables are relatively straight forward with one exception. Court-watchers frame decisions as either affirming or reversing a lower court's decision. However, the Supreme Court is sometimes the first instance and thus the original court of jurisdiction. To build a general model, the authors coded the Justice vote as either reversed, affirmed, or other. For their features, they tried 3 different types of features which may be useful in our own modeling:

- Encode categorical variables into binary or indicator variables
- Engineer new features, especially some related to Circuit Court of Appeals as those have been shown to be a strong predictor of reversal during certain periods in time.
- Create features that summarize the "behavior" of a Justice, the Court, the lower court, and the differences between them

Note that since the authors use a random forest classifier, they do not need to rescale, rotate, or remove features while for a neural net model we likely will have to do so. Additionally, the

authors trained 5 trees per term and added that information to the forest, enabling them to decrease simulation times and have more stable predictions as only a small number of trees are trained as they add a year of information from their starting point. This demonstrates their adherence to out-of-sample applicability and figuring out how we should group years to follow that for a neural network will be important.

Other researchers offer glimpses as to how best to do feature engineering by presenting additional features in their work. In A Computational Analysis of Oral Argument in the Supreme Court, Dickinson (2018) points out that, “human observers can take cues from the tone of a Justice’s voice and infer likely leanings from the phrasing of a tricky question or an imbalance in questions directed to one side or the other”. Unfortunately, given computers lack the ability to interpret these sentiments within arguments like human observers and at baseline lack the background case law knowledge, features must be created to allow for robust prediction. Dickinson’s research focuses on what we can model. Particularly his research extracts and uses the following question count features: Number of Questions to Petitioner or Respondent; Average Words Per Question to Petitioner or Respondent; Percent Questions to Petitioner or Respondent. The more questions a judge asks to a party, the more likely the judge is to vote against that party[1].

Additionally, Dickinson (2018) found that judges might direct more and longer questions to the side they disfavor to expose weaknesses and so created additional features on question chronology: First Question to Petitioner or Respondent; Average Consecutive Questions to Petitioner or Respondent. First Question features indicate how many questions into the argument a particular Justice asked his first question. Average Consecutive Questions features indicate the average number of consecutive questions a Justice asked a party without another Justice interrupting to ask a question. The rationale for these measurements stems from “the scarcity of oral argument time and the Court’s reputation for vigorous questioning, a Justice’s unusually sustained or especially early questioning may signal that the Justice has serious doubts about a party’s case”. In order to get at the tone that humans can hear, sentiment features were added to calculate Average Sentiment Towards Petitioner or Respondent. Utilizing all these features, the authors were able to achieve a 70% predictive accuracy with an SVM model for many justices, which they then used to predict the final outcome.

Aside from Dickinson’s research, Black et al. (2011) show that the number of “pleasant” versus “unpleasant” words in questioning at oral argument correlates with case outcomes, with a party’s probability of winning jumping by about 20% when the Court directs more unpleasant language to his or her opponent[2]. It is important to consider the whole sentence in pleasant vs unpleasant words. For example, when there’s a negation before a positive sentence. This paper

[1] Johnson et al., supra note 7, at 156–59; EPSTEIN ET AL., supra note 3, at 317–24; see also John G. Roberts, Jr., Oral Advocacy and the Re-emergence of a Supreme Court Bar, 30 J. SUP. CT. HIST. 68, 75 (2005).

[2] Black et al., supra note 9, at 577 (finding correlation between question pleasantness and case outcomes over 1979 to 2008 study period).

showed that the CoreNLP annotator works better in this regard than a dictionary model based on a bag of words. For our sentiment analysis, we use a pre-trained transformer-based model.

The literature is not just focused on predicting case outcomes, though. For example, Hawes (2009) ran multiple experiments with various models looking at case outcomes, but also whether the outcome is liberal / conservative, how Justice Thomas specifically will vote, or which exchanges between justices are most informative. Of their four experiments, they found a LIBSVM 2.86 implementation of an SVD with 5-fold validation works best for forecasting outcomes and the political leaning of the outcome while their other experiments used decision trees / tables and found no strong findings. In terms of case outcome classification, Hawes explores the relationship between conversational interactions (our dataset) and case variables, reaching an accuracy ranging from 62.5% to 76.8%. The results on Justice Thomas are overall inconclusive, but they do suggest that even justices with little contribution to the oral arguments in court are at least partly influenced by the discussion. One item to flag from this study, however, is that they utilized cases only from February 2006-April 2008 to have a consistent judiciary, which goes against our principal of generalizability from the Katz et al. (2017) study that we are striving for. This also led to their sparseness of data and caused them to want to add sentiment analysis in addition to bag-of-words and discourse n-grams that they used.

In sum, this paper will focus on the outcome forecasting task within Medvedeva et al.'s (2022) framework. Previous literature evaluates the predictive power of Supreme Court oral arguments on case outcomes as well as justices' votes with varying levels of success. Overall, the evidence suggests that the discussion matter, as well as conversational components are potentially useful predictors for this task. Existing literature has leveraged some NLP techniques to train machine learning models including random forests and SVMs. However, we find little exploration of more advanced natural language models that can better grasp the nuances of oral conversation. Thus, our goal in this project is to expand existing literature by exploring the power of more complex pre-trained models in predicting Supreme Court case outcomes and judges' votes.

Data and ETL

To obtain the data, we download the different datasets following convokit instructions[3]. Two raw datasets are provided: a case dataset with overall case information, and a utterance dataset with specific details for the different utterances of each case. For the overall case information, we get nested dictionaries that we then flatten so that we can have 1 row per case. This means for some cases we have ~20 advocates listed while for others we have much fewer (2-3). While this results in many blank features for each row, this is something we can easily filter if necessary. From there, we select relevant features and drop the remaining ones. This process can be found in "flattener_with_filter.ipynb" and "case_info_relevant_cols_only.csv".

We remove unnecessary metadata from the utterances and wrangle it into a practical format. Mainly, this includes removing whitespace from the utterances themselves and creating columns that keep track of who is addressed by a speaker. Since we know who a speaker replies to, we can also deduce who is being addressed which is additional information our models will be able to leverage.

We then join our overall case data to each utterance by case_id into the final dataset from which various models can select features. The data can be found in our data directory. The contents of the final files can be found below:

Feature	**Type**	**Definition**
id	string	Utterance id composed of a (different) case id and a sequence reflecting the other of speech
text	string	Single utterance
speaker	string	Person that said the utterance If it starts with "j_" it is a justice speaking
reply_to	string	Id of utterance being replied to
conversation_id	int	Identifier of the conversation; each case can have multiple conversations if there are multiple hearings
case_id	string	Unique case id
speaker_type	string	Type of speaker for the utterance ● A: Advocate ● J: Justice

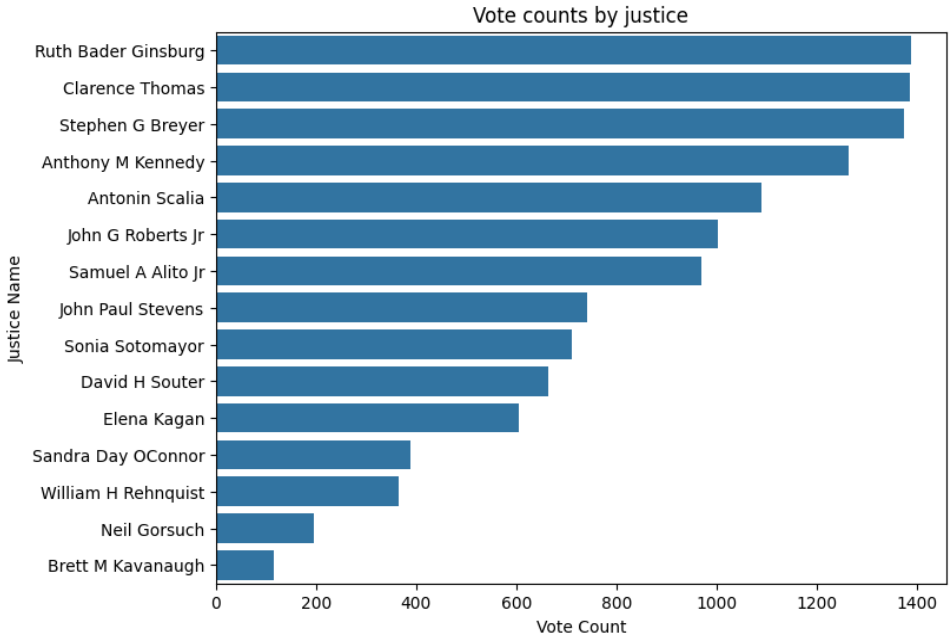
[3] See https://convokit.cornell.edu/documentation/supreme.html

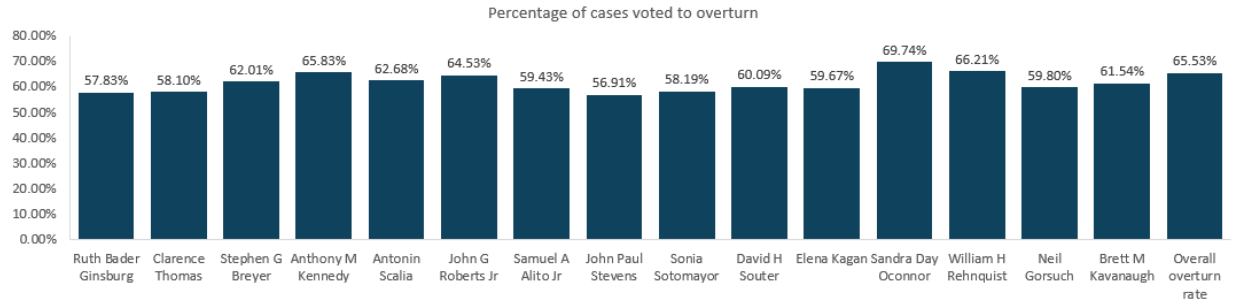
		• None: Not classified
side	Int	Side of speaker • 0: Respondent • 1: Petitioner • 2: Amicus curiae • 3: Unknown • None: Not classified
timestamp	float	Time speaker began speaking
addressing	string	Id of next utterance (who speaking to)
year	int	Year of case
title	string	Case title
petitioner	string	Petitioner name
respondent	string	Respondent name
adv_sides_inferred	bool	Informs if heuristics were used to fill in at least one advocate's side • True: Heuristics used • False: Advocates sides are all from explicit transcript mentions
known_respondent_adv	bool	If the respondent's advocate is known • True: Respondent advocate is known • False: Respondent advocate unknown
win_side	int	Who won the case • 0: Respondent won • 1: Petitioner won • 2: Unclear on who won
is_eq_divided	bool	If the vote was equally divided • True: Means vote was equally divided; cannot determine who the justice voted for • False: Means vote was not equally divided and can confirm who the justice voted for
votes_side_<justice>	int	Gives information about a specific justice's votes • -1: No information • 1: Dissented • 2: Voted with majority

		• None: Did not vote on case
advocates_<number>_<id>	str	Who the advocates in a given case are where number is what number lawyer they are and id is a unique identifier for the lawyer
advocates_<number>_side	int	Which side the advocate in a given case argues for • 0: Respondent • 1: Petitioner • 2: Amicus curiae • 3: Unknown • None: Unknown or inaudible speaker
speaker_replied_to	str	Speaking replying to in this utterance • A: Advocate • J: Justice • None: Not classified
speaker_type_replied_to	str	Type of speaker for the utterance • A: Advocate • J: Justice • None: Not classified
speaker_adddress	str	Speaker addressing in this utterance
speaker_type_addressed	str	Type of speaker being addressed in this utterance • A: Advocate • J: Justice • None: Not classified

Exploratory Data Analysis

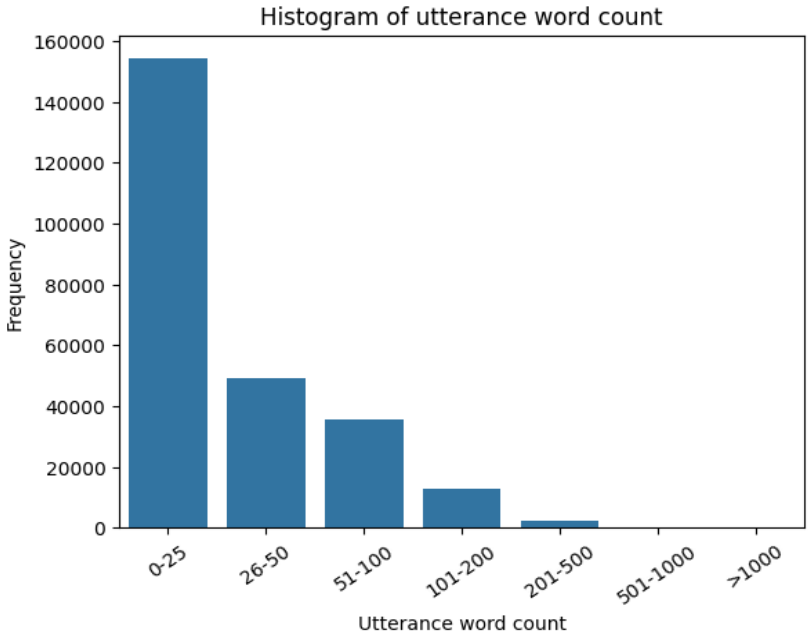
In the exploratory data analysis section, we take a deeper look into our Supreme Court dataset. We begin with an investigation of vote counts and overturn rate for each justice, providing a quantitative understanding of their voting patterns. Next, we turn our focus to the court utterances, creating a histogram that displays the frequency of different utterance word counts, giving insight into the length and complexity of typical court dialogues. Additionally, we examine the frequency of utterances per case through another histogram, shedding light on the volume of discourse in each case. Lastly, we create a histogram of the median utterance word count for each case, allowing us to discern the typical length of a court dialogue within individual cases. This exploratory analysis provides key information for our predictive modeling, giving insights into the data's underlying patterns and potential predictors.



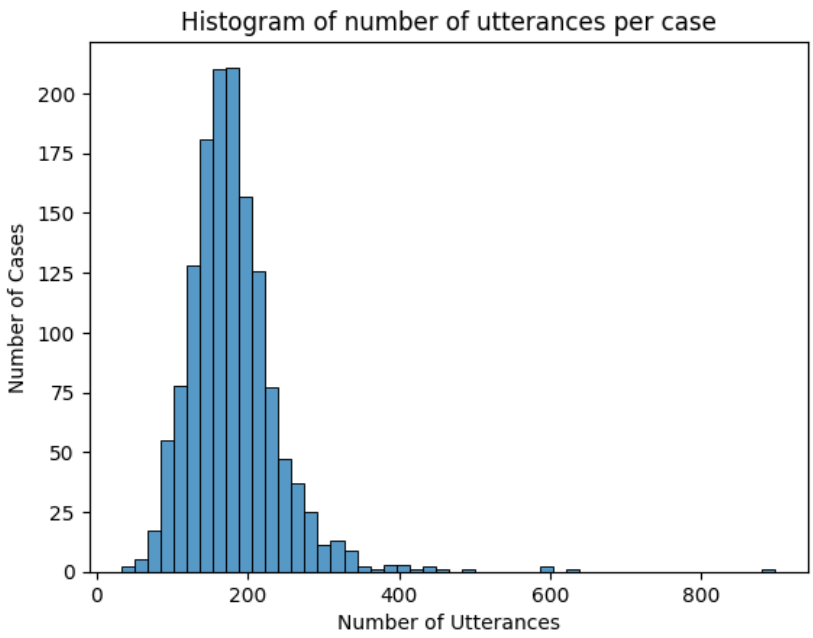


Here we see the number of times each justice has voted in a case and what percentage of the time they overturn the prior court ruling. Ruth Bader Ginsberd has the most cases where the newly instated Brett Kavanaugh has the fewest. In terms of predictive potential, we might

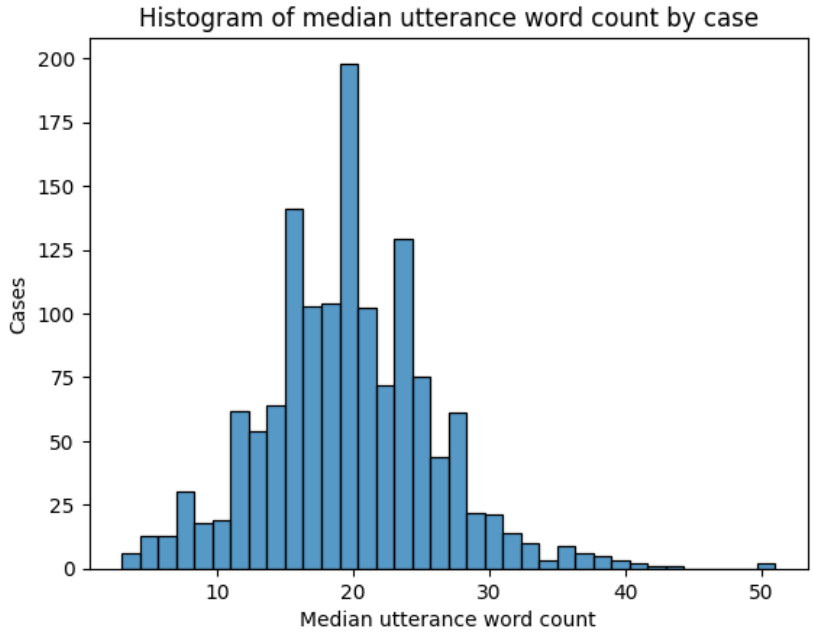
hypothesize that for the judges with more cases, our models will perform better as we have more data. One potential concern here is that when we predict at the case level, there are only around 1400 observations. Additionally, some justices vote to overturn much more frequently (such as Sandra Day O'Connor) - these justices may prove harder to find improvements on than a simple naive prediction.



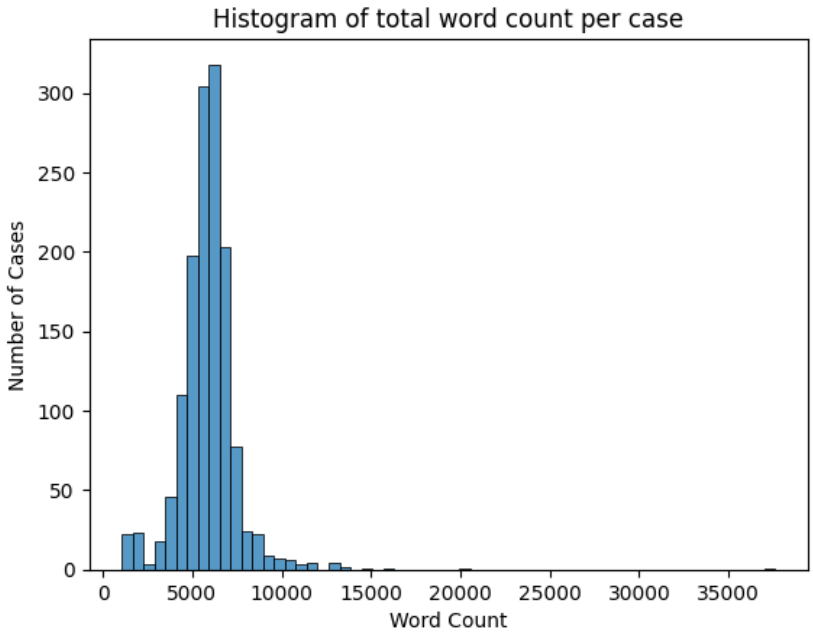
This plot shows us a histogram of word count of the utterances. Our lowest bucket 0-25 words houses the majority of the utterances. This is a low amount of information per utterance. Very few utterances are above 200 words.



This histogram shows us the total number of utterances per case. The distribution seems to hover right around 200 words skewing to the right with some outliers over 400 words.



When we look to median word count per case, the distribution evens out, though it is still skewed somewhat to the right.



Since a lot of our models make more sense looking at textual data at the case level, we concatenate data to the case level. This is especially useful with our pre-trained hugging face models. Here is a histogram showing how many total words we see in the concatenated case-level data. Again, it skews to the right. Overall, the middle of distribution sits around 6500 words per case.

Methodology

This project aims to understand the predictive power of Supreme Court oral arguments to forecast case outcomes, as well as justice votes. We have implemented both analyses as binary classification processes: aiming to predict 1 if the petitioner wins or the justice votes in favor of the petitioner, and 0 otherwise. For this report, we have used the most recent years of available data: 2000-2019. Future analyses may look at the changes in predictive power over time, as Supreme Court justices change and legislation evolves. We have built vote predictive models for all fifteen justices that served during this time period (order by most cases adjudicated):

- Ruth Bader Ginsburg (1993-2020)
- Clarence Thomas (1991 - present)
- Stephen G. Breyer (1994 - 2022)
- Anthony M. Kennedy (1988 - 2018)
- Antonin Scalia (1986 - 2016)
- John G. Roberts Jr. (2005 - present)
- Samuel A. Alito Jr. (2006 - present)
- John Paul Stevens (1975 -2010)
- Sonia Sotomayor (2009 - present)
- Elena Kagan (2010 - present)
- Sandra Day O'Connor (1981 - 2006)
- WIlliam H. Rehnquist (1986 - 2005)
- Neil Gorsuch (2017 - present)
- Brett M. Kavanaugh (2018 - present)

To train and evaluate each predictive model explained below, we divide the data into training, validation, and test datasets. Following our literature review, Katz et al. (2017) point out the importance on avoiding overfit on training data to reduce the influence of external factors - the validation dataset allows us to choose the best training parameters and evaluate results on our test dataset with no data leakage.

Sentiment Model

Previous literature suggests that justice sentiment can be useful in predicting case outcomes (Dickinson, 2018; Black et al., 2011). Based on this, we test the hypothesis by leveraging an NLP sentiment analysis approach. We explore a simple model that tries to predict the outcome of a case based on the sentiment of the justice utterances when addressing parties (petitioners and respondents). To estimate the sentiment of an utterance we use PySentimiento, a transformed-based library for NLP tasks. This model assigns to each utterance positive, neutral, and negative sentiment probabilities.

We first predict the outcome for both parties separately using the weighted ratio of positive versus negative sentiment probability. We evaluate different weights to choose the weighted ratio that maximizes our model performance. Our results are not promising. None of the weighted ratios returns a better performance than using a majority baseline.

Our next step is to compare the sentiment of justices towards petitioners with the sentiment towards respondents, instead of evaluating the average sentiment of the utterances by case for each party separately. For each case, we retrieve what party had the highest ratio of positive versus negative sentiment probabilities and compare it with the outcome of the case. Again, this method did not outperform the majority baseline.

In general, using overall justice sentiment does not allow us to predict the outcome of the cases. The academic literature that we reviewed shows that models previously built can perform well with certain justices while offering poor performance for other justices. We retrieve what party had the highest ratio of positive vs. negative sentiment probabilities and compare it with the outcome of the case (see Results section).

Bag of Words Neural Networks

Given the limited power of sentiment analysis to predict case outcomes and votes, we look at a series of models that can better leverage the oral arguments information for our classification task. The first models we implement are Neural Networks with Bag-of-Words (BoW) embedding. This embedding process simplifies the text data into a collection of words (tokens) disregarding the order and structure of the text. We first test a simple count-based preprocessing, creating the vocabulary from the top words appearing on our training data. This is a similar approach to that explored in the literature review. To improve this model, we develop continuous BoW using GloVe (Global Vectors for Word Representation), which captures meaning and semantic relationships between words in a high-dimensional space. These two models can not only provide information on the topics discussed, but they can also identify positive versus negative language, building on our prior sentiment analysis.

In order for these models to learn, we train them at the utterance level, rather than case level, which gives us a higher volume of observations. The training/validation/test dataset split is done at case level to avoid data leakage. We train neural networks with two hidden layers, a ReLU activation function after the first one and Sigmoid for the final output. We fine-tune parameters including vocabulary size for the count-based models, layer dimension, and learning rate. We use an Adam optimizer and a Binary Cross Entropy loss function. We use validation log loss to select the best number of epochs and a ROC curve for the best threshold.

Note that while the words used during the oral arguments can provide insights into the conversation, and the GloVe embedding can add some extra context, there is significant information missing in the input to these models. For instance, there is no information on the structure of the arguments, who was the speaker for each utterance, who they were responding to, and the flow of the conversation. The literature review does suggest conversational parameters such as the number of questions, interruptions, addresses, etc. is partly predictive of case outcomes (Dickinson, 2018; Hawes, 2009). The transformers models detailed below aim to build on these additional insights to develop predictions.

Pre-Trained Natural Language Models

Natural language models, including the transformer-based models specified below outperform BoW models due to their contextual understanding and semantic representations. These models can better capture relationships between words and phrases, considering the surrounding of words and their order, which allows to grasp the meaning and nuances of the text more accurately. Natural language models also excel at handling lexical and syntactic ambiguity by inferring meaning from context, which may be particularly useful for oral arguments.

For the preprocessing of the pre-trained language models, we concatenate all the strings from the oral arguments for each case. We include utterance start and end tags, as well as additional information about the speaker and the party being addressed. For instance, if Justice Gingsburg is speaking to the respondent, the new string would be:

'<UTTERANCE_START> Justice Gingsburg says "utterance" to respondent <UTTERANCE_END>'.
This approach ensures that our models can effectively utilize the information within an utterance by considering the context of the speaker and the party being addressed.

With this data we run the models below.

BERT

BERT (Bidirectional Encoder Representations from Transformers), as its name suggests, is a bidirectional transformer. Unlike recent language representation models, BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications. (Devlin, J. et al., 2018)

BERT being pretrained and fine-tunable as well as having a good performance track record, made it really attractive to our NLP task. The only adaptation we had to make for the model to run was to change the data so that a case is represented by a single row. As outlined above, this row contained all of the outcomes of interest and a single text that reflected all the utterances of the case. Thus, we concatenated the text of each utterance of a case by adding some information about who was speaking and who they were addressing.

However, there was one significant challenge as BERT only takes a maximum of 512 tokens but each case had much more than that amount. We addressed this by selecting a random starting point from which we would select the next 3000 characters for each case. While this means that we did not utilize all the information about a case, it allowed our models to be less computationally greedy which was extremely helpful as it was rather challenging to run our models without running out of GPUs and/or memory.

RoBERTa

RoBERTa (Robustly Optimized BERT approach) is a state-of-the-art natural language processing (NLP) model developed by Facebook AI and based on the BERT (Bidirectional Encoder Representations from Transformers) architecture. It builds upon the success of BERT and incorporates several improvements to enhance its performance RoBERTa is a transformer-based model that learns contextual representations of words and sentences. One significant improvement in RoBERTa is the modification of BERT's training methodology.

RoBERTa is trained on a larger corpus of unlabeled text data compared to BERT, encompassing a wide range of domains and genres. This extensive pretraining helps RoBERTa develop a more robust understanding of language and improves its ability to generalize across different tasks and domains. Another enhancement in RoBERTa is the removal of BERT's next sentence prediction (NSP) task during pretraining. BERT was pretrained using a masked language model (MLM) objective and an NSP objective, where the model predicted whether two sentences in a pair were consecutive or randomly swapped. RoBERTa removes the NSP task, enabling it to focus solely on the MLM objective. This change allows RoBERTa to better learn contextual representations and improve its understanding of sentences. (Liu, Y., Ott, M., Goyal, N., et al., 2020).

Both RoBERTa and BERT have limitations when it comes to the number of tokens they can handle. We ran two RoBERTa models: the first model selected the first 512 tokens in order, while the second model randomly chose a starting point and selected the next 3000 characters for each instance. Our next models remove the 512 token limitation, which makes them more suitable for the task.

BigBird

Transformers-based models, such as BERT, have been one of the most successful deep learning models for NLP. However, one of their core limitations is the quadratic dependency (mainly in terms of memory) on the sequence length due to their full attention mechanism. To remedy this, BigBird utilizes a sparse attention mechanism that reduces this quadratic dependency to linear. BigBird can handle sequences of length up to 8x of what was previously possible using similar hardware (Zaheer, Manzil et. al, 2020).

In addition to sparse attention, BigBird also applies global attention as well as random attention to the input sequence. Theoretically, it has been shown that applying sparse, global, and random attention approximates full attention, while being computationally much more efficient for longer sequences. As a consequence of the capability to handle longer context, BigBird has shown improved performance on various long document NLP tasks, such as question answering and summarization, compared to BERT or RoBERTa (Big Bird Hugging Face).

LongFormer

Recognizing the advancements made by BigBird, the advent of Longformer takes it one step further, offering an approach to tackling the long-range dependencies issue prevalent in large

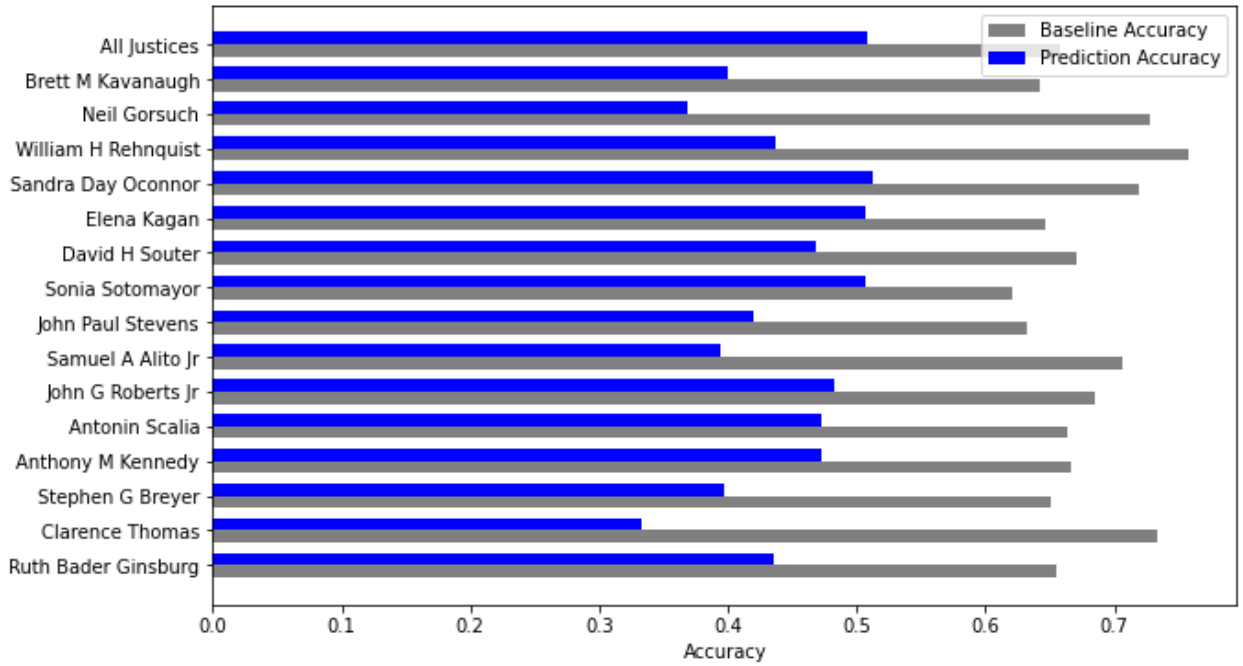
text bodies. Like BERT, RoBERTa, and BigBird, Longformer is a transformers-based model, but it differentiates itself by introducing a linear self-attention mechanism. Longformer uses a dilated sliding window attention, along with global attention, which means that it can capture both long-range dependencies and local context. However, it may still have limitations in capturing very long dependencies spanning multiple windows.

Longformer's proficiency in processing lengthier texts makes it an ideal candidate for tasks such as document summarization and question answering, which demand thorough analysis of extensive text. Like BigBird, its ability to overcome the token limitations experienced with BERT and RoBERTa allowed us to utilize the entirety of our data without sacrificing the richness of context in our text analysis tasks. Moreover, it alleviated some computational burdens, as the model no longer had to be excessively selective about token input due to the upper limit constraints (Beltagy et. al).

Results

Sentiment Model

Prediction accuracy:



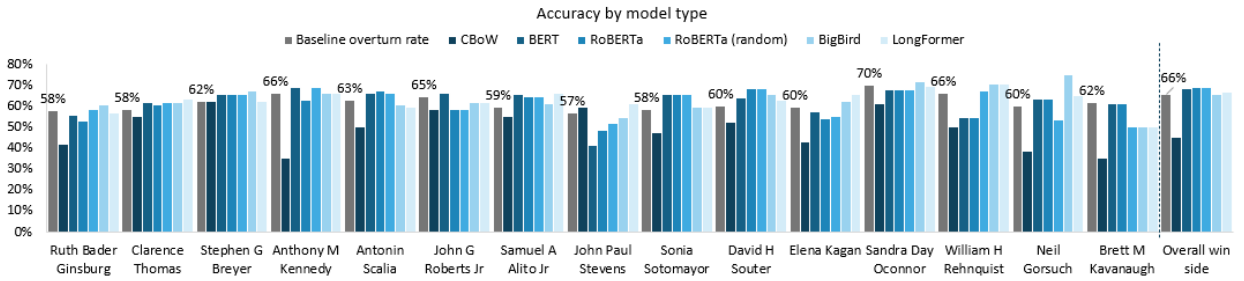
Our sentiment model, which utilizes a pre-trained BERT model built on various NLP tasks, did not surpass the majority baseline for any justice in terms of accuracy. These results indicate that the sentiment expressed in an utterance does not seem to be a reliable predictor of case outcomes, at least not in isolation. A future iteration of this work could manually classify the utterances to then train/fine tune a sentiment model on this new classified sentiment dataset.

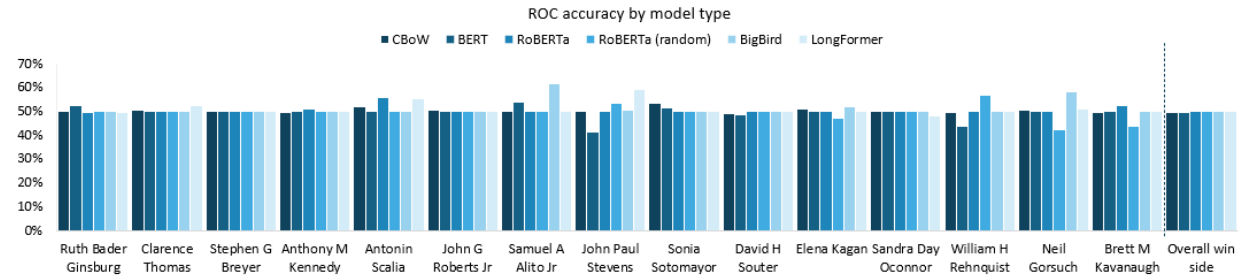
Remaining Models

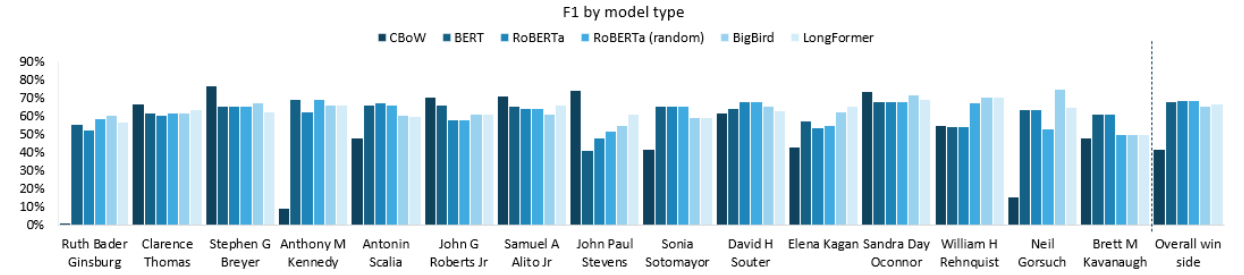
Find below the results obtained from the different models implemented (see the appendix for the F1 score and ROC AUC score tables).

Prediction accuracy:

	Baseline overturn rate	*CBoW*	*BERT*	*RoBERTa*	*RoBERTa (random)*	*BigBird*	*LongFormer*
Ruth Bader Ginsburg	57.83%	41.82%	55.50%	52.63%	58.37%	60.43%	56.83%
Clarence Thomas	58.10%	54.92%	61.72%	60.77%	61.72%	61.87%	63.31%
Stephen G Breyer	62.01%	62.00%	65.70%	65.70%	65.70%	67.39%	62.32%
Anthony M Kennedy	65.83%	35.11%	68.95%	62.63%	68.95%	66.14%	66.14%
Antonin Scalia	62.68%	50.28%	65.85%	67.07%	65.85%	60.55%	59.63%
John G Roberts Jr	64.53%	58.48%	66.23%	58.28%	58.28%	61.39%	61.39%
Samuel A Alito Jr	59.43%	54.97%	65.75%	64.38%	64.38%	60.82%	65.98%
John Paul Stevens	56.91%	59.19%	41.07%	48.21%	51.79%	54.67%	61.33%
Sonia Sotomayor	58.19%	47.11%	65.42%	65.42%	65.42%	59.15%	59.15%
David H Souter	60.09%	52.17%	64.00%	68.00%	68.00%	65.67%	62.69%
Elena Kagan	59.67%	43.04%	57.14%	53.85%	54.95%	62.30%	65.57%
Sandra Day Oconnor	69.74%	61.16%	67.80%	67.80%	67.80%	71.79%	69.23%
William H Rehnquist	66.21%	50.15%	54.55%	54.55%	67.27%	70.27%	70.27%
Neil Gorsuch	59.80%	38.33%	63.33%	63.33%	53.33%	75.00%	65.00%
Brett M Kavanaugh	61.54%	35.13%	61.11%	61.11%	50.00%	50.00%	50.00%
Overall win side	**65.53%**	**45.03%**	**68.10%**	**68.57%**	**68.57%**	**65.71%**	**66.43%**







As expected, the **continuous BoW** model generally performs poorly compared to the transformer models. We see a clear difference in the case outcome prediction, where accuracy is significantly lower than the baseline (45.03% vs. 65.53%). Similar results follow for most of the justices' vote predictions. As noted in the methodology, this model ignores information on the order and structure of the text, who is saying what, who is speaking to whom, etc. The results suggest that this information is of significant importance in predicting votes and case outcomes. Furthermore, by implementing this model at the utterance level in order to have enough observations, the model is overfitting on training cases with more utterances. This is particularly dangerous with a BoW model, where words specific to a case and unrelated to others may gain particular significance in the training process, which would result in poor evaluation accuracy. Moreover, due to time constraints we have implemented a pre-trained continuous BoW embedding process, not specifically trained for the task at hand. Further improvements may involve further training of the GloVe vectorization.

With a 61.61% average accuracy over the justices, **BERT** performs the second best out of all models in this category, just behind LongFormer (62.59%). Yet, it is only minimally above the baseline that is 61.51%. This can be attributed to many factors such as the token limit of BERT which certainly prevented us from achieving a higher accuracy. Furthermore, despite our

model's shortcomings, it was able to outperform the baseline for 10 of the 15 justices in our dataset. Furthermore, the number of cases present in the data for a given justice does not seem to be correlated with the performance of our model. In fact, the three justices for whom our model outperformed the baseline by the largest margin were the justices with the 9th, 7th, and 10th most cases respectively while the baseline outperforms us for the judge with the most cases. This is surprising as we expected to perform worse on the justices that ruled on less cases as we have less data for them. Finally, when predicting which side the court would side with on a given case, BERT performed better than on individual justices with an accuracy of 68.10%, 2.57% above the baseline.

Although **RoBERTa** was anticipated to enhance BERT's performance by leveraging its improved understanding of contextual representations, the actual improvement observed is minimal for some justices and absent for others. RoBERTa performed the best on predicting the outcome of cases at an accuracy rate of 68.57%, an improvement over the baseline of 3 percentage points. Surprisingly, the RoBERTa model that randomly selected a starting point for token retrieval did not outperform the RoBERTa model that simply obtained the first 512 tokens. Comparing both models yields mixed results, with no clear indication of a superior model.

As explained in the methodology section, **BigBird** removes the 512 tokens limitation and extends the BERT architecture by introducing sparse attention mechanisms to handle long-range dependencies more efficiently. This allowed us to feed the complete oral arguments into the models, including a lot more information to learn from. As a result, BigBird accuracy outperformed the baseline and other models in a majority of justices, with an average accuracy of 63.16% per justice (versus a baseline of 61.51%). However, the improvements are not very significant compared to other models, in fact BERT and RoBERTa outperform BigBird for the overall case outcome by almost 3 percentage points. While handling longer sequences efficiently, BigBird may sacrifice fine-grained modeling of local context since it uses larger blocks and reduced attention on local tokens. The observed results may indicate that global context is more significant to predict justice votes, while local nuances are more predictive of case outcomes.

In contrast to BigBird, **Longformer**'s sliding window attention aims to balance the capture of long-range dependencies and local context. The results of Longformer align with the previously identified patterns. Compared to BigBird, we see a slight increase in accuracy for case outcomes (66.43%), but lower performance at the justice vote level with an average of 62.59% per justice. We see the opposite effect when comparing Longformer results with BERT and RoBERTa, which have an even higher focus on short dependencies. It is important to note we had considerable time and computational constraints that conditioned our ability to fine-tune parameters for all models implemented. We suspect that further fine-tuning of hyperparameters, including even longer sequences than those considered in our implementation, may better reflect the benefits of BigBird and LongFormer over BERT and RoBERTa given the long sequence nature of our task.

It is important to note that the majority of models show an ROC around 50%, which means that they perform at chance level or exhibit random guessing behavior. This suggests that the models are not effectively capturing the underlying patterns or relationships in the data to make accurate predictions. The ROC value of 50% indicates an equal chance of correct and incorrect predictions, indicating a lack of discriminatory power. We find very few instances where models yield ROC above 50%. For example, BigBird for Samuel A Alito (61.81%) Jr and Long Former for John Paul Stevens (59.11%). In the next steps section, we highlight a few suggestions to improve the presented results.

Discussion

Our exploration into using various neural net NLP models for predicting Supreme Court outcomes and justices' votes has produced mixed results with a lot of potential for future work. As our literature review confirms, predicting case outcomes is a very difficult modeling task, and the few existing models only predict a few percentage points higher than a baseline rate (always predicting petitioner). As our bag of words model underperformed relative to our transformer based pre-trained models, this reinforces the importance of structural information and important informational context that the more advanced architectures can make use of.

We sought to predict both individual justice votes and the overall outcome of the case. Our transformer models, specifically RoBERTa and BigBird, performed the best. RoBERTa , despite having a token limit of 512, performed the best on predicting the outcome of cases at an accuracy rate of 68.57%. This is an improvement over the baseline of 3.04%. Longformer follows closely at 62.59% When predicting the individual votes of justices, on average our BigBird model performs the best with an average accuracy of 63.16%. BigBird outperformed the baseline for 12 of the 15 justices.

Surprisingly, the performance of our models did not correlate with the number of cases available for each justice, indicating that the quality of data may play a more significant role than the quantity. Another interesting thing to note is that the models didn't seem to predict better for specific judges, this suggests that we aren't able to conclusively determine if certain judges are more swayed by oral arguments.

Next Steps

In terms of next steps, there is significant development that can be tested to improve the results presented on this report. In this section we make multiple suggestions for future development.

First, further research could investigate training a sentiment model using a manually classified dataset. The dataset should include labels that describe the sentiment of each utterance. By combining this approach with the utilization of a pre-trained model, the next step would involve fine-tuning the model using the newly classified dataset. This process would enhance the model's suitability for the sentiment analysis task at hand.

Second, the continuous BoW model implemented uses pre-trained embeddings which may not capture the legal nuances and specificities of the task at hand. As an improvement, we could look at training context-specific embeddings or utilizing pre-trained embeddings specific to the legal sector. This could provide the model with additional complexities for this type of language and increase predictive power.

Next, as the results for the BoW model highlight, it is very important to capture as much context as possible when confronted with modeling problems like ours. Our literature review suggests that information other than the oral arguments provided is also likely to impact case outcomes,

including public pressures, circuit court appeals, etc. (Katz et al., 2017). Therefore, we would recommend improving data nuance by capturing more detailed information, such as the interaction dynamics between the speakers in the courtroom, explicitly inputting information such as our sentiment analysis into the models, and potentially adding more variables additional information on the case. All of this additional context may provide the information required for improved predictions.

Furthermore, due to computational and time restraints, we were not able to perform an in depth grid search on our hyper parameters. It is common practice to implement such a grid search to obtain the model with the best predictive power. Ensuring optimized hyperparameter fine-tuning may help not only boost results, but also better discern the relative performance of different model types.

Workload Management

We took a divide and conquer approach on various aspects of the paper, with each team member reading papers and/or writing the summaries of the literature review. Leading up to the mid-quarter evaluation, there was ample time spent on EDA and constructing helper functions / bag of words model with the intent of using the helper functions for larger models. Unfortunately, we ran into difficulty on using helper functions in Collab. To try and ease the workload, Jonas and Sergio worked on some of the data cleaning scripts necessary for our larger models and then Matt helped streamline some scripts for individual justice predictions alongside Nuria and Michael. Most of the time spent on this part of the project was spent learning how to better utilize the HuggingFace library and tuning which parameters gave the best results (e.g. learning rate / epochs). Overall, we worked collaboratively and supported each other in each individual task.

A more specific division of tasks can be found below:

Jonas Heim
- Baselines
- Data cleaning
- BERT modeling
- RoBERTa modeling

Matt Kaufmann
- BigBird modeling
- LongFormer modeling
- Poster

Michael Wagner
- Exploratory Data Analysis
- Visualizations
- BigBird modeling

Núria Adell Raventós
- CBoW modeling
- BigBird modeling
- Poster

Sergio Olalla Ubierna
- Sentiment Modeling
- BERT modeling
- RoBERTa modeling

Bibliography

Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

BigBird, huggingface.co/docs/transformers/model_doc/big_bird. Accessed 19 May 2023.

ConvoKit: A Toolkit for the Analysis of Conversations. Jonathan P. Chang, Caleb Chiam, Liye Fu, Andrew Wang, Justine Zhang, Cristian Danescu-Niculescu-Mizil. Proceedings of SIGDIAL. 2020.

Black, R. C., Treul, S. A., Johnson, T. R., & Goldman, J. (2011). Emotions, oral arguments, and Supreme Court decision making. The Journal of Politics, 73(2), 572-581.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, A. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arxiv.org/abs/1810.04805.

Dickinson, G. M. (2018). A Computational Analysis of Oral Argument in the Supreme Court. Cornell JL & Pub. Pol'y, 28, 449.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Echoes of power: Language effects and power differences in social interaction. Cristian Danescu-Niculescu-Mizil, Bo Pang, Lillian Lee and Jon Kleinberg. WWW 2012.

Hawes, 2009. Computational Analysis of the Conversational Dynamics of the United States Supreme Court. https://drum.lib.umd.edu/handle/1903/9999

Katz DM, Bommarito MJ II, Blackman J (2017) A general approach for predicting the behavior of the Supreme Court of the United States. PloS One 12(4):e0174698

Medvedeva, M., Wieling, M. & Vols, M. Rethinking the field of automatic prediction of court decisions. *Artif Intell Law* **31**, 195–212 (2023). https://doi.org/10.1007/s10506-021-09306-3

Roberts Jr, J. G. (2005). Oral Advocacy and the Re-emergence of a Supreme Court Bar. Journal of Supreme Court History, 30(1), 68-81.

Sharma RD, Mittal S, Tripathi S, Acharya S (2015) Using modern neural networks to predict the decisions of Supreme Court of the United States with state-of-the-art accuracy. In: International conference on neural information processing. Springer, pp 475–483

Zaheer, Mazil et al. (2020). Big Bird: Transformers for Longer Sequences. Neural Information Processing Systems (NeurIPS) 2020. https://proceedings.neurips.cc/paper/2020/file/c8512d142a2d849725f31a9a7a361ab9-P

aper.pd decisions of Supreme Court of the United States with state-of-the-art accuracy. In: International conference on neural information processing. Springer, pp 475–483

Appendix

Results - F1 Score:

	Baseline overturn rate	CBoW	BERT	RoBERTa	RoBERTa (random)	BigBird	Long-Former	Average improvement
Ruth Bader Ginsburg	57.83%	1.30%	55.50%	52.63%	58.37%	60.43%	56.83%	-10.32%
Clarence Thomas	58.10%	66.63%	61.72%	60.77%	61.72%	61.87%	63.31%	4.57%
Stephen G Breyer	62.01%	76.52%	65.70%	65.70%	65.70%	67.39%	62.32%	5.21%
Anthony M Kennedy	65.83%	9.51%	68.95%	62.63%	68.95%	66.14%	66.14%	-8.78%
Antonin Scalia	62.68%	48.37%	65.85%	67.07%	65.85%	60.55%	59.63%	-1.46%
John G Roberts Jr	64.53%	70.63%	66.23%	58.28%	58.28%	61.39%	61.39%	-1.83%
Samuel A Alito Jr	59.43%	70.87%	65.75%	64.38%	64.38%	60.82%	65.98%	5.94%
John Paul Stevens	56.91%	74.29%	41.07%	48.21%	51.79%	54.67%	61.33%	-1.69%
Sonia Sotomayor	58.19%	41.76%	65.42%	65.42%	65.42%	59.15%	59.15%	1.19%
David H Souter	60.09%	61.95%	64.00%	68.00%	68.00%	65.67%	62.69%	4.96%
Elena Kagan	59.67%	43.06%	57.14%	53.85%	54.95%	62.30%	65.57%	-3.53%
Sandra Day Oconnor	69.74%	73.59%	67.80%	67.80%	67.80%	71.79%	69.23%	-0.08%
William H Rehnquist	66.21%	54.98%	54.55%	54.55%	67.27%	70.27%	70.27%	-4.23%
Neil Gorsuch	59.80%	15.95%	63.33%	63.33%	53.33%	75.00%	65.00%	-3.81%
Brett M Kavanaugh	61.54%	48.09%	61.11%	61.11%	50.00%	50.00%	50.00%	-8.15%
Overall win side	65.53%	41.99%	68.10%	68.57%	68.57%	65.71%	66.43%	-2.30%

Results - AUC Score:

	Baseline overturn rate	CBoW	BERT	RoBERTa	RoBERTa (random)	BigBird	LongFormer
Ruth Bader Ginsburg	57.83%	50.12%	52.60%	49.66%	50.00%	50.00%	49.74%
Clarence Thomas	58.10%	50.37%	50.00%	49.94%	50.00%	50.00%	52.25%
Stephen G Breyer	62.01%	50.06%	50.00%	50.00%	50.00%	50.00%	50.00%
Anthony M Kennedy	65.83%	49.66%	50.00%	51.01%	50.00%	50.00%	50.00%
Antonin Scalia	62.68%	52.01%	50.00%	55.65%	50.00%	50.00%	55.32%
John G Roberts Jr	64.53%	50.66%	50.00%	50.00%	50.00%	50.00%	50.00%
Samuel A Alito Jr	59.43%	50.03%	54.07%	50.00%	50.00%	61.81%	50.00%
John Paul Stevens	56.91%	49.92%	41.12%	49.94%	53.38%	50.50%	59.11%
Sonia Sotomayor	58.19%	53.38%	51.27%	50.00%	50.00%	50.00%	50.00%
David H Souter	60.09%	48.82%	48.71%	50.00%	50.00%	50.00%	50.00%
Elena Kagan	59.67%	51.04%	50.00%	49.82%	46.89%	52.02%	50.00%
Sandra Day Oconnor	69.74%	50.11%	50.00%	50.00%	50.00%	50.00%	48.21%
William H Rehnquist	66.21%	49.67%	43.75%	50.00%	56.67%	50.00%	50.00%
Neil Gorsuch	59.80%	50.54%	50.00%	50.00%	42.11%	58.33%	51.19%
Brett M Kavanaugh	61.54%	49.36%	50.00%	52.60%	43.51%	50.00%	50.00%
Overall win side	65.53%	49.52%	49.65%	50.00%	50.00%	50.00%	50.00%