Predicting Bill Party Sponsorship in New York State Congress between 2008 and 2022 using Natural Language Processing

Intro

Motivation

The motivation of our project derives from using Natural Language Processing on the given title and summary of a bill in the New York State Congress between 2008 and 2022, to predict whether or not its sponsor is Republican or Democrat. As an example, the title of Assembly Bill A3726 presented in 2009 is "Establishes the environmental sustainability education act" and the summary is "Recodifies the animal cruelty laws from the agriculture and markets law to the penal law and reclassifies certain penalties within; repealer". So we may ask:

Can we use Natural Language Processing on the summary and title of bills in New York State Congress between 2008 and 2022, to predict whether or not a bill's sponsor is Republican or Democrat?

Most past research on bills is determining the chance that a bill will become law. For example, Matthew Hutson explains his findings stating "Sponsors in the majority and sponsors who served many terms were at an advantage when predicting the bill's success" (2017). Additionally, while most studies focus on factors including chamber size, sponsor demographics, and other similar predictors when looking at bills, our objective is to build on this research by specifically observing the relationship between a bill's syntactic content within its title and summary and the party of the sponsor. For example, there may be different sentence length, word use, or content between the summary and title of a Republican's bill versus a Democrat's.

While our research mainly focuses on building two models, one on the title and another on the summary of a bill, we use their predictions along with other text-based variables to create a more accurate model. These variables include the length of the summary and title, as well as 6 keywords, which we in our understanding of New York State politics, believe may be

used more often by one political party versus the other. These words are: "Crime", "Fund", "Education", "Health", "Free", and "Environment". We assume that Democrats may be more likely to present bills regarding topics of education, the healthcare system, and the environment, whereas Republicans may be more likely to present bills regarding topics of criminality, funds, and freedom.

A problem within political research is that the majority occurs on the Federal level, while State legislators and legislation "increasingly hold the keys to the civil rights of America's citizens" (2012). Specifically, New York State is a hotbed for legislation that many other states follow, so we must hold our legislators accountable. The goal of this model is to create a tool for constituents and political organizations in New York State to determine whether or not a representative aligns with the ideals of their stated party. For example, if we predict a bill by a Democratic sponsor to be sponsored by a Republican based on the summary and title of their bill, this may be a red flag for activists to hold their representative or the specific bill accountable.

Data Source

The data source that we are using is the Open Legislation v2.0 API Docs. This API provides up-to-date New York State Congressional information including Bills and Resolutions, Committee Calendars, Member Information, and much more. The specific API that we implement is the Bills and Resolutions API. Using this API one can retrieve plentiful bill information with different specifications, from retrieving information on a specific bill to searching for a list via keyword.

We are currently implementing the Bills and Resolutions API by first downloading general information on all bills and resolutions for New York State Legislative Sessions: 2008-2009, 2010-2012, 2013-2014, 2015-2016, 2017-2018, 2019-2020, and 2021-2022 as JSON files, converting these into dataframes, and then binding the dataframes to each other row-wise. Each row represents a unique bill. Next, we choose the variables that we consider most important and filter the chamber that the bills appeared in to Senate and Assembly. Note that we downloaded all bills for the legislative sessions included in the API as of April 21st, 2023.

While the only information that we need to build our models include the bill's sponsor, summary, and title, we chose to extract other important information, to build a more robust dataset for future research. The variables we extracted include the bill's unique ID, the final printed bill ID (this may append A, B, C, D, or E to the bill's unique ID), the first year of the legislative session that the bill was introduced in, the congressional chamber the bill was introduced in, the bill's title, the date and time the bill was published, the full name of the bill's sponsor, the bill's summary, whether the bill was adopted in New York State (it passed senate, assembly, and was signed by the Governor), and the last date a decision was made on the status of the bill in Congress.

As bills' sponsors' political parties are not included in the legislative API, we gathered this information from an outside source. Using Ballotpedia, we attempted to scrape the political party of the 452 unique sponsors in our 14-year congressional timeline, and for the sponsors that we could not scrape, we conducted internet searches to find their political party. We then merged this party data, with 1 representing Democrat and 0 representing Republican, with our primary dataframe.

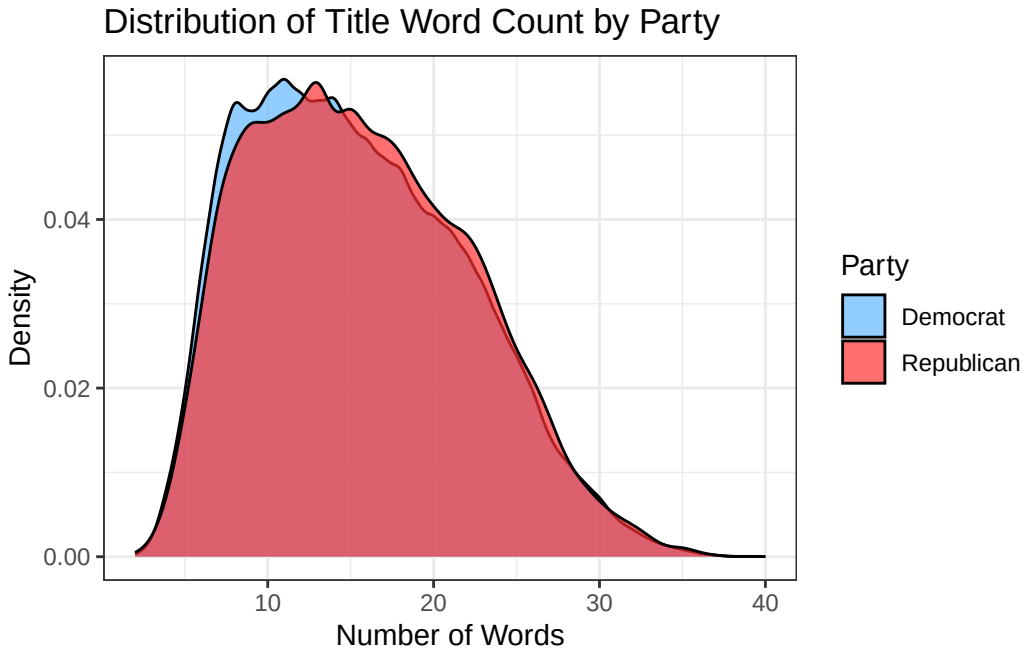
Finally, we created the dataframe that we would use for prediction by choosing the summary, title, and party columns from the dataframe described above. Additionally, we created two additional columns for the number of words in the summary and the title. We then created 6 more columns that counted the number of times each of the 6 keywords appeared within the summary text. This left us with a final dataframe of 11 variables: the summary, title, party, length of summary, length of title, the number of times "Crime" was used in the bill's summary, the number of times "Fund" was used in the bill's summary, the number of times "Education" was used in the bill's summary, the number of times "Health" was used in the bill's summary, the number of times "Free" was used in the bill's summary, and the number of times "Environment" was used in the bill's summary, as well as 132289 observations (unique bills).

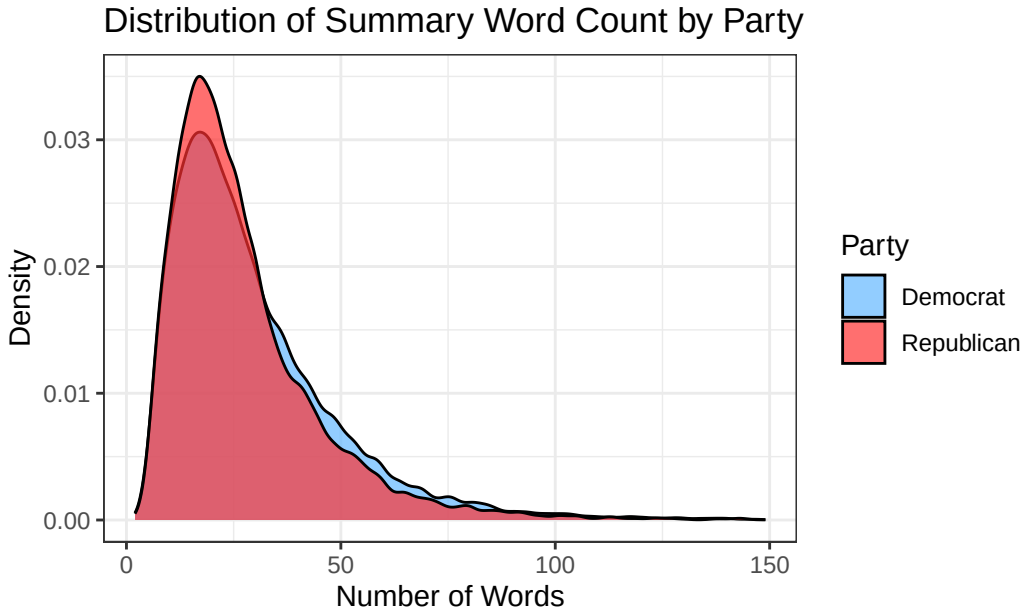
Methodology

Data Exploration

Before we fit our model, we conducted preliminary data exploration. We found that in our dataset, 95,042 (71.84%) bills were sponsored by Democrats, and the other 37,247 (28.16%) were sponsored by Republicans. This means that our variable to predict is unbalanced, with the Democrat category being much more prevalent than the Republican category, and so we need to be careful to not over-predict Democrats. We also found that over the 14 years, 294 (65.04%) congress members were Democrats and the other 158 (35.96%) were Republicans. Together, these statistics tell us that while Democrats only made up 65.04% of all congress members in New York State during this period they sponsored 71.84% of bills. This means that Democrats are generally more likely to sponsor bills than Republicans.

We also found that the summary and titles for Democrat-sponsored bills had an average of 30.00 and 15.34 words respectively, whereas the summary and titles for Republican-sponsored bills had an average of 27.61 and 15.69 words. The plot below shows the distribution of the number of words by each party.



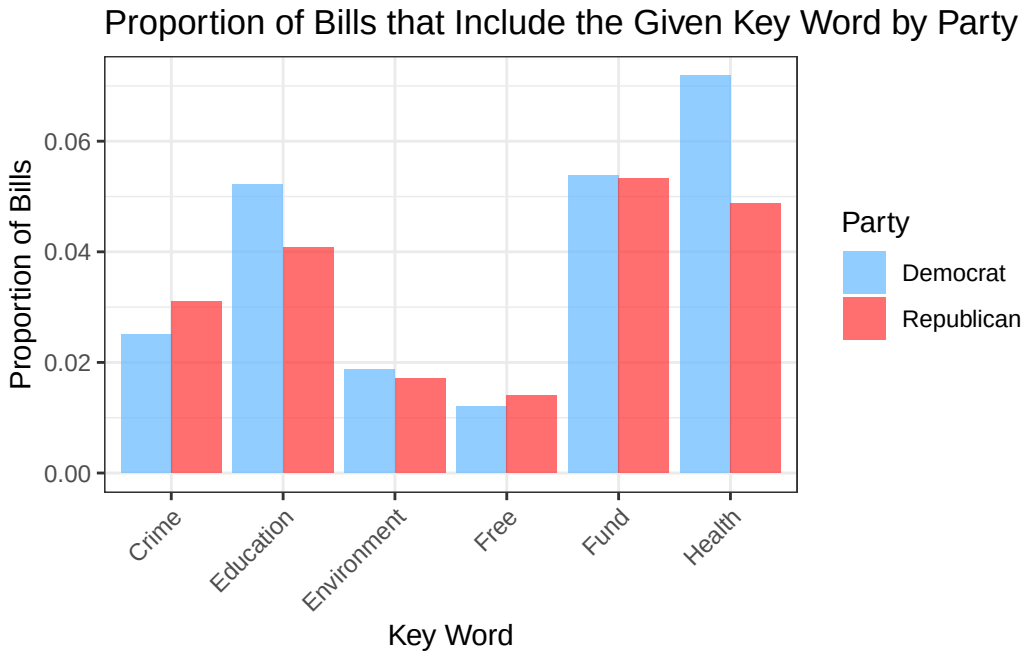


This contains summaries with less than 150 words for graphical purposes

These plots show us that the number of words in a summary or title may not be a good predictor of a sponsor political party. As we can see, the distribution of the number of words in the summary and title of a bill along with the means are almost identical for both political parties. The only noteworthy difference is in the distribution of summary word count, we see that Republicans have a much higher density at their mean than Democrats. Even though

they appear to be quite similar, proceeding with the project, we will include these variables to ensure we use the maximum amount of data.

Lastly, we looked at the proportion of occurrences of our keywords in bills. As a reminder, the words that we chose include "Crime", "Fund", "Education", "Health", "Free", and "Environment". The plot below shows the proportion of bills whose summaries include the given keyword by sponsor party.



Model Building

We begin our predictive task by splitting our data into a random training, validation, and test set, where the training set includes 60% of the data, the validation set includes 30%, and the test set includes 10%. Thus the training set has 79339 observations, the validation set has 39669 observations, and the test set has 13224 observations. We create the training set to fit the summary and text models on. Next, we use the predictions from those models on the validation set as variables for the full model to be fit on. We then fit the full model on the validation data and check our final predictive accuracy on the test data. We split into 60% training in order to have a sufficiently large training set. We also need a large validation set as this is used to fit the final model, so we chose 30%.

First, we fit two vectorization layers that processes our summary and title text data to convert them into numerical representations that can be used as input to our models. In these layers, we specify the maximum number of unique words to be considered during the tokenization process as 500. If there are more than 500 unique words in the text data, the words that occur

less frequently will be ignored. We also specify the maximum length of the sequence of tokens using Cross Validation for both vectorization layers. The tokenization process itself works by splitting each string into substrings of words, recombining the words into tokens, indexing the tokens as integers 1 through 500, and then turning the index of tokens into a vector whose length is chosen through Cross Validation, using the integer representations. These vectors are then stacked on each other to create a matrix representation of the data.

Next, we fit our text classification models. First, we specify the input shape of the data. In our case, we consider a single string as input. The next layer takes the string input and tokenizes it into a sequence of integers using the vectorization layer we described above. Next, we create dense embeddings of the integer sequence, which capture the semantic meaning of the words in the input text. Then, we fit a Bidirectional Recurrent Neural Network that processes the sequence in both directions, which helps to capture long-term dependencies in the input text. We chose to fit Bidirectionally because this allows us to use future context to learn more about the single words. The Long Short-Term Memory layer has 32 units, which determine the complexity of the model. Next, we have a dropout layer which randomly drops out some of the nodes in the previous layer during training, which helps to prevent overfitting. We then have a 256 unit dense layer which is a linear transformation to the output of the previous layer and applies the ReLu activation function to generate a prediction. We use the ReLu activation function because it allows the model to learn nonlinear relationships between the input features and the output predictions. Next, we fit another dropout layer to again help against overfitting. Finally, the last layer in the model is a sigma activation layer. This layer generates probabilities between 0 and 1 for our models of being a Democrat. In both of models, the maximum length of the sequences, the embedding size, and the dropout rate are all chosen through Cross Validation.

We compile the model using the Adam optimization algorithm as it is generally preferred over normal stochastic gradient descent in Natural Language Processing. The Adam optimization algorithm is similar to gradient decent but it takes "into consideration the 'exponentially weighted average' of the gradients" allowing "algorithm converge towards the minima in a faster pace.". Additionally, we use binary cross entropy for our loss function as this allows us to make accurate binary predictions.

After fitting our summary and title models, we use their prediction on the validation data, along with the lengths of the summary and title columns and the key word columns, all standardized, as predictors in a final full model. The first layer of the final model is a dense layer that uses a ReLu activation function. The second layer is a drop out layer, to prevent overfitting. The next layer is another ReLu activation function. Next, the fourth layer is another drop out layer. The output layer has a single output unit using a sigmoid activation function, which outputs a value between 0 and 1 representing the predicted probability of the input belonging the Democrat class. The number of nodes in the dense layers and both drop out rates are chosen via Cross Validation.

Summary Model

We fit 5-fold Grid Search Cross Validation on the summary model to find the best hyper-parameters. The parameters that we include are the maximum length of 50 or 100 words, an embedding size of 25 or 50, and dropout rates of 0.2 or 0.5. We score the best model in terms of AUC. We use AUC as our metric rather than accuracy because we are predicting an unbalanced variable. AUC is a good metric for unbalanced prediction tasks because it takes into account both classes equally, without being dependent on class size. We fit the Cross Validation with 2 epochs each and a batch size of 128, for the sake of computational speed.

Fitting the Grid Search Cross Validation, we find that the best summary model in terms of AUC is the model with a maximum length of 100 words, an embedding size of 50, and a dropout rate of 0.2 with an AUC of 0.679. We then fit the summary model with the best parameters using 30 epochs and a batch size of 32 to get precise results.

Title Model

We fit the same 5-fold Cross Validation for our title data, with the same parameters, batch size, and number of epochs.

This Grid Search Cross Validation shows us that the most accurate model built on the titles has a maximum length of 100, an embedding size of 50, and a dropout rate of 0.2. The AUC of this model is 0.679. We then fit this model on the training data using 30 epochs and a batch size of 32.

Full Model

Although both models are good predictors of a bill sponsorship's party, we can leverage them to build a stronger model. For example, if for a certain bill, the summary model gave it a prediction of 0.75 but the title model gave it a prediction of 0.25, what should a final full model predict- a Republican or a Democrat? To answer this question, we calculate predictions for the validation data using the summary and title models and use these as predictors in a final Neural Network. Additionally, we increase the predictive power of this model by adding the summary lengths and keyword predictors to this dataframe that we predict. Note that since our predictors have different ranges, for example, our prediction variables are between 0 and 1, but our lengths and word counts can be any positive integer, we standardize them.

We use 5-fold Grid Search Cross Validation to find the best model. The Cross validation parameters are a first drop out rate of 0.25, 0.5, or 0.75, a second drop out rate of 0.25, 0.5, or 0.75, a dense layer of sizes 64, 128, or 256, and a second dense layer of sizes 64, 128, or 256. We fit this data using a batch size of 32 and 2 epochs like the other Cross Validation models. Fitting our Grid Search Cross Validation, we find that the best model in terms of AUC is the model with a dense 64-node layer, then a drop out rate of 0.5, a second dense layer of 128

nodes, and then a drop out rate of 0.75, with an AUC of 0.792. We fit a model with these parameters using 100 epochs and a batch size of 32 to get our final model.

Results

To determine the overall strength of our models, we used the models' AUCs, along with their confusion matrices and accuracy scores at the 0.5 thresholds and at the threshold where sensitivity and specificity determined by the ROC Curve are equal. We do this with the goal of finding a model that predicts Democrats and Republican sponsors equally well. To get a graphical metric to understand our predictions, we also plot the distributions of the predictions using density plots.

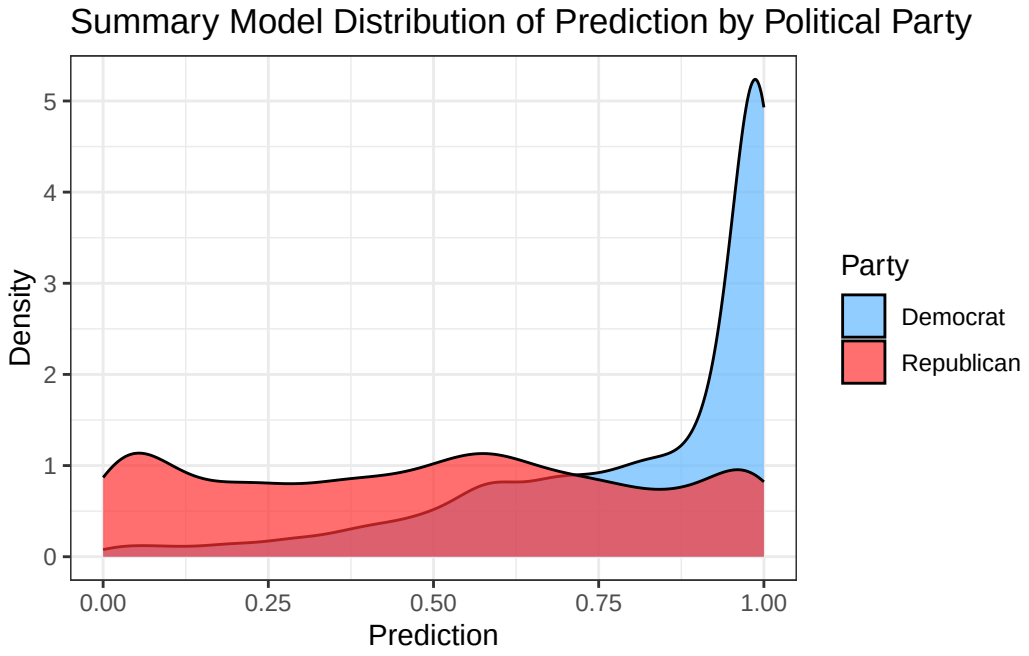
Summary Model

The summary model fit on the test data gives an AUC of 0.795. Below are the confusion matrices for the model using a 0.5 threshold and a threshold where specificity is equal to sensitivity of 0.715.

Table 1

	Threshold = 0.500		Threshold = 0.715	
	Predicted Dem.	Predict Rep.	Predicted Dem.	Predicted Rep.
True Dem.	8411	1047	6815	2643
True Rep.	1920	1846	1052	2714

Using a threshold of 0.5, we get an accuracy of 0.776. Our confusion matrix tells us that the model predicted 88.93 percent of Democrat-sponsored bills correctly and 49.018 percent of Republican bills correctly. Using the threshold where specificity is equal to sensitivity of 0.715, we get an accuracy of 0.721. The confusion matrix for this adjusted threshold tells us that the model predicted 72.055 percent of Democrat-sponsored bills correctly and 72.066 percent of Republican bills correctly. Below is a plot of the distribution of predictions.



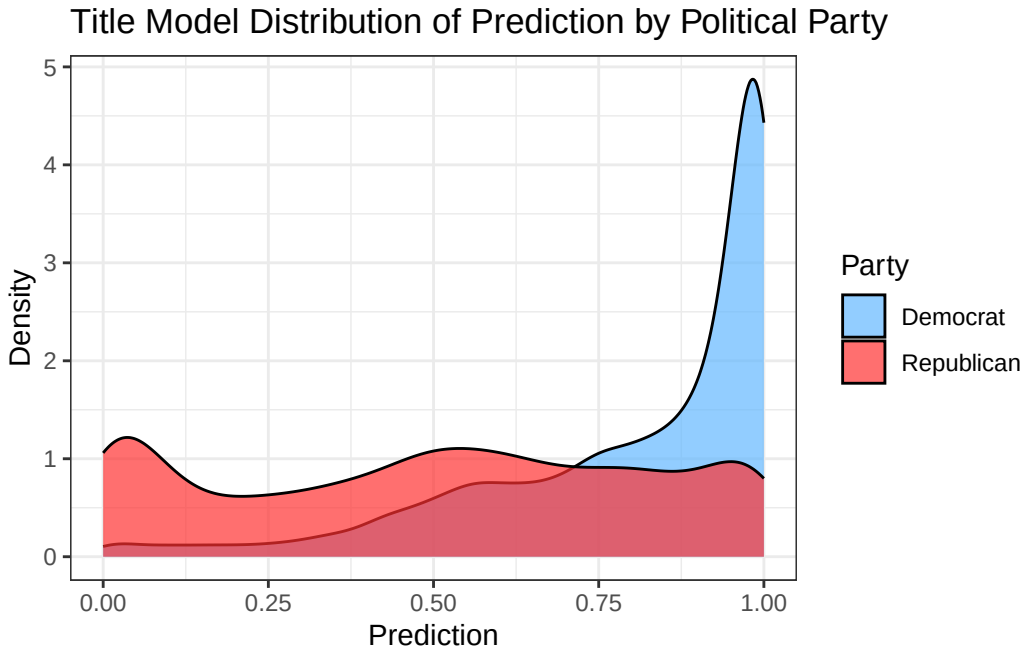
Title Model

The title model data gave an AUC of 0.783. Below are the confusion matrices for the model using a 0.5 threshold and a threshold where specificity is equal to sensitivity of 0.731.

Table 2

	Threshold = 0.500		Threshold = 0.731	
	Predicted Dem.	Predict Rep.	Predicted Dem.	Predicted Rep.
True Dem.	8404	1054	6747	2711
True Rep.	1980	1786	1079	2687

Using a threshold of 0.5, we get an accuracy of 0.771. The confusion matrix tells us the model predicted 88.856 percent of Democrat-sponsored bills correctly and 47.424 percent of Republican bills correctly. Using the threshold where specificity is equal to sensitivity of 0.731, we get an accuracy of 0.713. The confusion matrix for this adjusted threshold tells us that the model predicted 71.336 percent of Democrat-sponsored bills correctly and 71.349 percent of Republican bills correctly. Below is a plot of the distribution of predictions.



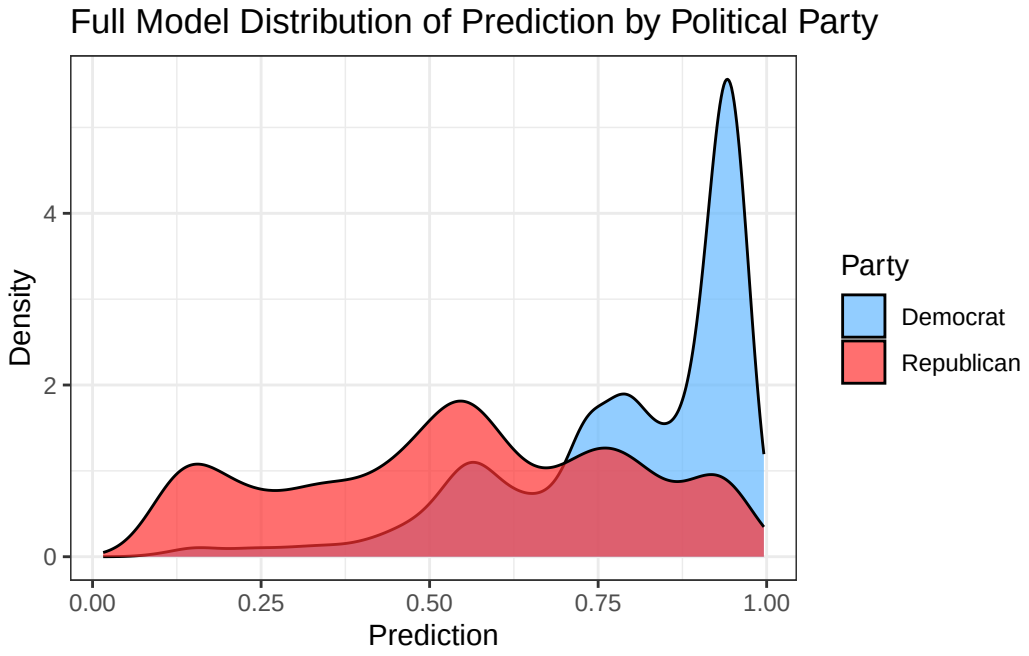
Full Model

The full model gave an AUC of 0.798. Below are the confusion matrices for the model using a 0.5 threshold and a threshold, where specificity is equal to sensitivity of 0.729.

Table 3

	Threshold = 0.500		Threshold = 0.729	
	Predicted Dem.	Predict Rep.	Predicted Dem.	Predicted Rep.
True Dem.	8404	1054	6762	2696
True Rep.	1980	1786	1088	2678

Using a threshold of 0.5, we get an accuracy of 0.771. The confusion matrix tells us that while the model predicted 88.856 percent of Democrat-sponsored bills correctly and 47.424 percent of Republican bills correctly. Using the threshold where specificity is equal to sensitivity of 0.729, we get an accuracy of 0.714. The confusion matrix with this threshold shows us that the model predicts 71.495 percent of Democrat-sponsored bills correctly and 71.11 percent of Republican bills correctly. Below is a plot of the distribution of predictions.



Discussion

Interpretation

Our results demonstrate that both the summary and title models have high AUCs, 0.795 and 0.783 respectively. This means that both models fit our data quite well. As the AUC of the summary model is higher than the AUC of the title model, this tells us that the summary model is better at predicting party of a bill's sponsor than the title model. Fitting both models with a threshold of 0.5, we see that the accuracy scores are quite high, at 0.776 and 0.771 respectively. This means that the summary model predicts bill sponsor parties correctly 77.564 percent of the time and that the title model predicts bill sponsor parties correctly 77.057 percent of the time. While these accuracies are quite high, they can be misleading.

We see from the confusion matrices at the 0.5 threshold that the summary model predicts 88.93 percent of Democrat sponsors correctly, but only 49.018 percent of Republican sponsors correctly. Similarly, the title model predicts 88.856 of Democrat sponsors correctly and 47.424 percent of Republican sponsors correctly. This indicates a vast over-prediction of Democrats.

To fix this problem, we use thresholds where specificity is equal to sensitivity. For the summary model this threshold is 0.715 and for the title model this threshold is 0.731. Doing this, we get accuracies of 0.721 for the summary model and 0.713 for the title model. Once again, this accuracy metric tells us that the summary model is better at predicting bill sponsorship parties than the title model in terms of accuracy. In fact, looking at the confusion matrix from

both models with the adjusted thresholds, we see that the summary model predicts 72.055 percent of Democrats and 72.066 percent of Republicans correctly, whereas the title model only predicts 71.336 percent of Democrats and 71.349 percent of Republican correctly. Since the summary model is able to predict a higher proportion of Democrats and Republicams correctly, this again tells us that the summary model is superior in our prediction task. Although the summary model predicted better than the title model in terms of AUCs, accuracy scores, and the confusion matrices, our metrics show us that both models were quite good at predicting sponsor parties.

Examining the distribution of the summary and title models, we can see that the distributions are nearly identical. This is because the summaries and titles are highly correlated. Comparing the small differences between the distribution for the summary and title models, we can see that the summary model distribution of Republican predictions has a drop around 0.2 that the title model does not have and similarly that the title model distribution of Republican predictions has a drop at 0.85 that the summary model does not have.

The ideal plot of prediction would be to have two separable distributions, one distribution centered near 0 for Republicans and one centered near 1 for Democrats. From both plots, we can see that the distribution for Democrats is very left-skewed with a maximum near 1, as we should hope. Both distributions also have two small plateaus, one starting at around 0.6 and one starting near 0.75. While the distribution for Democrats is what we hoped for, the distribution for Republicans is far from what we wanted. Instead of being centered near 0 and right-skewed, the distribution is almost flat, with three humps near 0.1, 0.55, and 0.9. This tells us that the models' predictions for Republicans are almost only slightly better than picking a random number between 0 and 1.

Using these same metrics, we can assess the predictive power of the full model and compare it to the summary and title model used to build it. The AUC of the full model is 0.798. This means that the full model fits the data very well and has the highest AUC out of all three models. Additionally, when we fit the model with a threshold of 0.5, we get an accuracy of 0.771. This means that the model correctly labels bill parties 77.057 percent of the time. However, from the confusion matrix we see that while the model predicts Democrat sponsors correctly 0.889 percent of the time, the model only predicts Republican sponsors correctly 47.424 percent of the time. These percentages are worse than the summary model but slightly better than the title model. Adjusting the threshold to 0.729, we get an accuracy score of 0.714 and that the model predicts Democrats and Republicans correctly 71.495 percent and 71.11 percent of the time, respectively.

The distribution of the full model is quite surprising. As is expected, the distribution of Democrat predictions is left-skewed, however, there are three noticeable maxima at 0.55, 0.8, and 0.9. Similarly, the distribution of Republican predictions appears to be centered near 0.5 and has four maxima at 0.15, 0.55, 0.8, and 0.9. This plot shows that the neural network used to build the full model seems to be amplifying the maxima in the summary and title plots. What this means in terms of the predictions themselves, is that the full model appears to be generating more predictions similar to the most prevalent predictions in the previous

two models. Overall, this distribution shows that neither the prediction for Democrats nor Republicans is very good.

In terms of AUC, the full model is superior with an AUC of 0.798, with the summary model being a close second with an AUC of 0.795. However, since we are interested in choosing the model that can be used to best classify Democrats as Democrats and Republicans and Republicans, the summary model with a threshold of 0.715 is the best, accurately predicting both Democrats and Republicans over 72% of the time, something which neither of the other two models achieve.

This tells us that going forward, activists, lawmakers, citizens, or anyone else can fairly accurately predict a bill's sponsor's political party simply by running the bill's summary through the summary model with a threshold of 0.715, providing a satisfying yes to our research question: **Can we use Natural Language Processing on the summary and title of bills in New York State Congress between 2008 and 2022, to predict whether or not a bill's sponsor is Republican or Democrat?**

Limitations

Some limitations we encountered in this project include computational space and time, as well as the generalizability of the data. Building our models was time intensive given the large size of our dataframe and our analysis of text data. As a result, we made decisions including limiting the number of epochs, increasing the batch size, fitting a model with few hidden layers and only checking a small number of hyper-parameters in Grid Search Cross-Validation. This means that there is likely a better model that exists and our model could be fit more and thus yield more predictive accuracy.

Other limitations come with the data itself. As our model was trained on New York State Congressional data between 2008 and 2022, we do not think it would be wise to attempt to use it to predict bill sponsor parties for a different state's bill summaries and titles. In another state, legislators may use a different vocabulary, grammar, and length of title and summary, making the model virtually unusable. Similarly, while we do believe that this model can and should be used for future bills in New York Congress, it is important to note that Democrat and Republican ideologies shift over time, so a bill that may have followed more of a Democratic ideology between 2008 and 2022 may fit in more with the Republican ideology in future years. This indicates that the model may incorrectly predict the bill sponsor as a Democrat.

In the future, we suggest that researchers limit the number of bills they use in their prediction, possibly to a single legislative session, and only predict Senate or Assembly. This would limit computational space and time and may provide more accurate predictions. Given less data, researchers may run models wherein they can fine-tune more parameters, to get better models. Additionally, due to time and computational constraints, we were unable to include the bills' actual text in the models. A bill's text is likely a much better indicator of the party of the

bill's sponsor than the summary and title. We suggest that researchers analyze the text, along with the summary and title in future research.

Overall, while we could have taken steps to improve our project, we are quite happy with its predictive capabilities and believe that it sufficiently answered the stated research question.

Sources

"Artificial Intelligence Can Predict Which Congressional Bills Will Pass." Science.org, 2021, www.science.org/content/article/artificial-intelligence-can-predict-which-congressional-bills-will-pass. Accessed 18 Apr. 2023.

"Ballotpedia." Ballotpedia.org, 2021, www.ballotpedia.org/Main_Page. Accessed 18 Apr. 2023.

"The Importance of State Legislatures, Redistricting, and Civil Rights." Brennan Center for Justice, 2021, www.brennancenter.org/our-work/analysis-opinion/importance-state-legislatures-redistricting-and-civil-rights. Accessed 24 Apr. 2023.

"Intuition of Adam Optimizer." GeeksforGeeks, GeeksforGeeks, 22 Oct. 2020, www.geeksforgeeks.org/intuition-of-adam-optimizer/. Accessed 24 Apr. 2023.

"Open Legislation V2.0 API Docs — Open Legislation 2.0-Alpha Documentation." Nysenate.gov, 2015, www.legislation.nysenate.gov/static/docs/html/index.html. Accessed 18 Apr. 2023.