Authorship Attribution of Supreme Court Opinions Using a Multi-Class Support Vector Machine

Michael Hazard
Tevon Strand-Brown
Kevin Chang
Computer Science
Stanford University

***Abstract*—Justices in the Supreme Court often submit opinions anonymously, or Per Curiam. This practice was created to express opinions when the verdict was clear, and generally unanimous. Per Curiam literally means 'of the court.' However, in recent years, Per Curiam opinions have been used to shield a justice's identity, particularly when writing a dissenting opinion. This presents a lack of transparency that is dangerous for our democracy. Through our research we sought to identify which justice penned any given opinion by training a model to identify their writing styles. We utilized a support vector machine (SVM) model trained on all prior opinions with an attributed author in the Supreme Court. We found that feature extraction which strikes a balance between thoroughly exploring and overfitting the data has the largest impact on our prediction accuracy. With the right set of features, we were able to reach satisfactory accuracy. Through this model, we are able to predict the correct justice (in a test set of never before seen data) nearly 3/4s of the time. We hope this type of research will lead to a more transparent, and trustworthy judiciary.**

***Keywords*—Supreme Court, Authorship Attribution, SVM, Feature Extraction, Stylometry.**

I. Introduction

The Supreme Court is a cornerstone of the US Government. Every year, the Court hands down around 75 decisions, each with comprehensive opinions detailing the logic of the verdict. Justices do not always write an opinion: some support majority opinions, while some write dissenting ones detailing why they disagree with the verdict reached by the court. The majority of these opinions are signed by the justice who authored them. However, in recent years, many more opinions are being handed down anonymously, or 'Per Curiam.' The practice of Per Curiam was created for opinions on cases that were 'open-and-shut' and where the court was in general consensus and would submit a single opinion. However there is a worrying trend of justices releasing these anonymous opinions in controversial cases, effectively allowing them to dissociate themselves from their anonymously published opinions (Robbins [1]). We hope continuing this research can help us maintain an open and transparent judiciary.

II. Literature Review

A. Authorship Attribution

Authorship attribution is a popular problem that researchers both inside and out of the Artificial Intelligence community have tackled. Machine learning techniques have been applied to cases like determining who wrote essays in the Federalist Papers, or attributing William Shakespeare's works (Merriam [3]). There are several types of features that have been highlighted and researched extensively to aid in authorship attribution. This at a high level can be broken into several categories: Function features (word usage and frequency), vocabulary features (diversity and type of vocabulary used), Sentence features (length, and composition of sentences) (Rudman [8]). N-grams have also been a popular way of featurizing textual information, as detailed by Wang [7]. They find that unigrams and bigrams are very effective in cases of textual attribution, however trigrams add little benefit. Our research reinforced this conclusion.

B. SVMs for Textual Attribution

SVMs have been shown to yield high accuracy when classifying text (Joachims [9]). Joachims demonstrates that they also well perform in high dimensional feature spaces, even with sparse data. Textual classification problems often are limited by the size of their corpus, as most authors have a limited body of work, and thus we see why there is significant use of SVMs for this type of research.

SVMs have also been shown to increase performance when exposed to virtual (created) data. (Manabu [10]) This is highly valuable in the case of textual classification as there may be limited availability of data, particularly for authorship attribution. This proved relevant to our research, as we were limited to the corpus of Supreme Court opinions. While we did not add synthetic data, research has shown that further work could benefit from an artificially expanded data set.

III. Opinions and Style

The writing of Supreme Court justices has been studied extensively since the court's founding in 1789. For example, one Yale Law set out with the goal "to identify and clarify the appellate judicial opinion as a distinct literary genre within the larger civic literature of the American republic of laws. (Ferguson [13]) Not only is the literary opinion now its own genre, but there have also been many studies on the tone of specific justices. "Roberts favors informal diction and a lightly ironic tone that tend to humanize otherwise dry institutional issues" (Raw [12]).

From this it follows that we can find features that identify specific justices.

IV. Features and Data

We gathered our data from Court Listener, a 501(c)(3) dedicated to keeping the court system transparent. They expose a bulk data API from which we retrieved all Supreme Court opinions, which we parsed to retrieve the plain text opinion and authorship information. This set contained Per Curiam opinions, which were separated from the authored opinions so we could train and test on labeled data.

Feature extraction is a key component of building a model. We used Li [5] as a reference point for developing our set of feature templates. We began by extracting unigrams, bigrams and trigrams. While these performed reasonably well, we were able to drastically increase our performance by refining these features. We began by removing trigrams, as they have not been shown to create better results (Wang [7]), and, in our model, added significantly to the size of our feature vector while decreasing our prediction accuracy. This is an example of a feature which caused the SVM to overfit the train set.

In addition to n-grams, we added a number of stylometric feature templates to help us discern the writing styles of particular justices. These included:

- Usage of punctuation (!, -, , , _, ?, ;).
- Sentence length
- Sentence openers (Next, therefore, lastly, etc.)

We also experimented with the Natural Language Processing (NLP) library Textblob to help us extract complex features from the opinions. These included:

- N-grams of root words, singularized and lower-cased (e.g. Geese $\rightarrow$ goose)
- Sentence composition (parts of speech and how they interact)
- Signed polarity of writing (sum of all sentence polarities).
- Absolute polarity (sum of absolute sentence polarities).
- Sentence subjectivity

Interestingly, many of these NLP feature templates were found to overfit the train set, and led to an increase in error on the test set.

Each feature template above was normalized in at least one way to remove the effect of variable opinion length. We normalized features by:

- Occurrences / number of words in opinion
- Occurrences / number of sentences in opinion
- Occurrences / average length in words of sentence in which feature appears

Lastly, we restricted the n-grams that we had extracted. We aimed to remove n-grams containing case specific (as in court case) language or jargon. To do this, we removed n-grams that did not appear in at least n cases. Iterative testing on values in the range $n \in \{5, 10, ..., 100\}$ found $n = 80$ to yield the best results. We also performed this same type of analysis on the upper end of n-gram frequency, imagining that the most common n-grams would be common between all justices; however, this decreased the model's accuracy.

After removing the least frequent (by case appearances) n-grams, we selectively added features back into our feature set by calculating:

$$F = \sum_{i=1}^{m} x_i * [y = j], j \in J$$

This is the total weight of a n-gram for a particular justice. We then calculated:

$$[max(f) > \alpha * \sum F/||F||], f \in F$$

This is the maximum usage of any justice. If this is more than α times greater than the average usage, we add that feature back into our feature set. We believe these high usage features indicate something unique about a particular justices writing style. Using this equation we found $\alpha = 1.4$ to be a strong value.

V. Addressing Skew

A challenge in addressing this problem was tackling the inherent skew in our data. The structure of the Supreme Court creates significant variation in the number of opinions written by each justice, largely due to the discrepancy in the length of each justice's career and the hierarchy of the court. The current Supreme Court is a perfect illustration of this skew, as Chief Justice Kennedy has written 186 opinions, while Justice Sotomayor has written 1 and justices Gorsuch and Kagan have written none.

To address this, when we stratified our training and test sets, insuring that the underlying distribution is the same throughout each set as across the entire corpus. We also utilized a penalized SVM, which heavily weights false-negatives on minority classes, preventing the model from simply learning the underlying distribution and guessing the most frequent class. We also investigated expanding our data set to include all legal opinions of each justice, as many serve long careers prior to receiving a position on the Supreme Court; however, this was not found to be tractable given our resources and the Court Listener API, as some cases were attributed to multiple judges. We also feared that this data would not improve the quality of the model as the length of previous careers was also significantly skewed, the previous opinions did not follow the same structure, and the data became significantly larger and more expensive to work with. To use this data one would have to selectively create a corpus with an equivalent representation of each judge, containing only opinions closely matching the syntax of those from the supreme court.

Further work to combat skew could include the implementation of an anomally detection model. The original SVM could be trained exclusively on majority classes, while a second SVM is trained on minority classes. The anomaly detection model would identify any given opinion that does not match one of the majority classes, and utilize the low count SVM to classify the opinion.

VI. SVM

We decided to use a SVM (Support Vector Machine) for several reasons: It is well studied in the area of authorship attribution, it handles large, sparse feature vectors well, and it handles small data sets very well as compared to other methods.

An SVM is a strictly binary classifier. Given a dataset $D = \{(x_1, y_1), ..., (x_m, y_m)\}$ of m training examples, where $x_i \in R^n$ represents the ith training data point and $y_i \in \{-1, 1\}$, the SVM uses the scoring function:

$$\sum_{i=1}^{m} \alpha_i y_i * K(x_i, x) + b$$

K here is the kernel function, which decides the shape of the decision boundary and will be discussed further. α_i is the coefficient associated with the ith training example, b is a scalar coefficient. Using this function, the SVM establishes a decision boundary between the samples belonging to each of the binary classes. However, the ideal decision boundary will not always separate the data completely, so a value C is introduced as the penalty for misclassifying a sample. Setting the value of C to be a large value leads to overfitting as it penalizes incorrect classification, and a low value of C can lead to a generalized model that does not predict with high accuracy. C is one of the hyper parameters that we examined extensively.

A. Choosing Hyper Parameters

We tested a matrix of hyper parameters that can be seen in Table 1. The key decision to make for hyper parameters is determining a kernel K. The kernel decides the shape of the decision boundary. The choices are linear, polynomial, and radial. For a linear kernel we only decide on C and the coefficient. For a polynomial kernel we also determine the degree of the polynomial. With a radial kernel, we choose a value for gamma, gamma controls the shape of the radial decision function. A small value of gamma leads to low bias and high valiance, and a high value of gamma leads to high bias but low variance.

Through grid-search of these hyper parameters we found a linear kernel with $C = 1$ to yield the best results.

VII. Multi-class SVM

The classification problem we are working on, however, is not binary. The Supreme Court usually has nine justices, thus we need to be able to classify justice opinions as one out of multiple justices. To do this, we constructed a multi-class SVM. Our multi-class SVM uses one-versus-one classification, meaning it constructs $n * (n - 1)/2$ binary SVM classifiers, where n is the number of labels in our data. Each of these binary classifiers independently 'votes' for the justice which it classified the example as, and the justice (label) with the most votes is classified as the prediction of the multi-class model.

VIII. Results

We experimented with a variety of tactics to reach our final result. In our first test, we extracted unigram and bigrams from the body of the opinion and trained a model on these, splitting data equally between training and testing sets. This yielded 48% accuracy. We found that changing our breakdown to 75% training and 25% testing data yielded better results, and thus we used this breakdown for the remainder of our tests.

Following this initial baseline we extracted further features and limited infrequent features as described above. Following these additions, our accuracy rose to 70%.

It was at this point that we realized that our data was not as clean as we expected (see Dirty Data). Following this discovery, we parsed all of the opinions and separated each majority opinion from the attached descents. Once the descents were separated, we removed opinions that had fewer than 500 words, as these were generally very short statements that had almost no suggestion of the justices particular writing style. With these additional modifications our average accuracy became the final 73.5%.

We validated this result by generating 20 separate tests. In each of these tests we would randomly split the data into training and test sets, stratified so that each held a representative sample of the data. We used 25% of data for testing. In each of these tests we trained a new model on the new training set, then tested its accuracy on the test set. 73% is that average accuracy of all of these tests, the results varied between 69-77%.

The tests yielded low training error when compared to the higher test error, which suggests that the has data has high variance. A model has the tendency to over fit the high variance data, and thus the only remedy to this issue is the addition of more data. This suggests that generative models may be the best next step as the Supreme Court cannot be coerced into seeing more cases. To this effect we are actively working to expand our data set for another rendition of this study.

In Figure 1, you can find a confusion matrix of our results. It shows the correct classification on the rows, and our predictions on the columns, and can provide insight into many of our assumptions about the current justices. Ginsberg and Scalia are known for having distinctive tons, Scalia is more effusive while Ginsberg has a drier to-the-point style. These distinctions are reflected in the data, as we classify them with very high accuracy.

The confusion matrix also highlights the areas that need the most improvement. With the help of a court expert, we may be able to identify features specific each justice, allowing us to improve accuracy on currently weak classes.

A. Trials

We tried a number of things that did not improve accuracy. We tried eliminating the most common features, and also the most common words. Both of these things decreased accuracy.

We also found that limiting word count of opinions to anything above 2000 words decreased our accuracy. We believe this is because it shrank the corpus, which was small to begin with.

IX. Dirty Data

Following the poster session we dove deeper into our data to try to extract additional impactful features. However what we found was that our data was not as clean as we had assumed. The author of a given opinion was classified to be the justice who had written the majority opinion. Sometimes, dissenting opinions (written by other justices) were included inside of the same opinion body and were thus labeled by the Court Listener API as being authored by the justice who wrote the majority opinion. Cleaning our data allowed our score hit the 73.5% mark.

Given our focus on classifying Per Curiam opinions, we think this granularity (of separating concurring and dissenting opinions) may help us even more in improving our accuracy. While it may not improve our test set accuracy (as these are all labeled opinions, not Per Curiam), it could improve our prediction of new data that is not labeled at all.

X. Future Work

A. Data

As discussed above, one of the major areas of focus for future work will be in gathering a more comprehensive data set. We hypothesize that a better implementation of the data gathered on a per justice basis from state and federal courts would help us address the skew in the Supreme Court dataset; however, this data must be examined more closely as it currently reduced the model's score. Furthermore, as has been mentioned before, SVMs have been found to improve given the addition of synthetic data, thus generative models may be fruitful.

B. Meaningful Features

While our manually selected features performed reasonably well, we believe that taking a more systematic or educated approach to feature selection could benefit our prediction accuracy.

One method to identify further features will be to use an LSTM (Long short-term memory) RNN on a paragraph level to extract further features on both the sentence and paragraph level. Hassan [11] found that LSTM RNNs performed extremely well on classification of sentences, this is because, unlike n-grams, they have a longer 'memory' than $n - 1$. This helps them retain semantic meaning and preserve word order, similar sentences could be clustered and used as additional features. This could highlight semantic tones for certain justices beyond the usage of identical words. We avoided this approach during the project as we feared that the exclusively Supreme Court data set was too limited to leverage this model.

Another method to enhance our feature set will be to consult with a Supreme Court expert. Before computers could preform reasonably well at stylometric tasks, a host of court observers would learn the relative styles of the justices in order to deduce their opinions. These court experts could help us identify particularly salient features that our model may not include, or may have included but had not recognized as important. This could help us identify features beyond the n-grams level, on the structural level of the text. This is something our model currently misses completely.

Through this combination of cutting edge machine learning, and human insight, we believe we can significantly improve the performance of our model.

XI. Conclusion

In this time where fact seems to be disputable, we believe that greater transparency is a positive thing. We hope to expand this research, with cleaner data and more professional input to create an even better predictor.

Our results were strong, but as we implement the details addressed in future work, we expect the accuracy of our model to continue to increase, especially with more data from pre-Supreme Court opinions.

If we were to repeat this project again, knowing what we do now, we would maintain our same work flow and pipeline. These worked extremely well and helped us iterate quickly on feature extraction, feature selection, testing and validation. However we would probably consider our initial data source more carefully, as we learned far along that additional data would be useful. Our established pipeline will allow us to quickly process data once a connection is built from a source to our pipeline, so we may be able to see improved results quickly!

In the meantime, we hope you have enjoyed reading this paper.

References

[1] Robbins,Ira, Scholarship highlight: The Supreme Courts misuse of per curiam opinions, SCOTUSblog (Oct. 5, 2012, 11:13 AM), http://www.scotusblog.com/2012/10/scholarship-highlight-the-supreme-courts-misuse-of-per-curiam-opinions/

[2] Moshe Koppel et al., Computational Methods in Authorship Attribution, 60(1) J. AM. SOCY INFO. SCIENCE TECH. 9 (2009).

[3] Thomas V.N. Merriam Robert A.J. Matthews, Neural Computation in Stylometry II: An Application to the Works of Shakespeare and Marlowe, 9(1) LITERARY LINGUISTIC COMPUTING 1 (1994).

[4] Elayidom, M Sudheep, et al. Text Classification for Authorship Attribution Analysis. Advanced Computing: An International Journal, vol. 4, no. 5, 2013, pp. 110., doi:10.5121/acij.2013.4501.

[5] Li, William, et al. Using Algorithmic Attribution Techniques To Determine Authorship in Unsigned Judicial Opinions. Stanford Technology Law Review, vol. 16 no. 3 2013.

[6] Stern, Henry, et al. Predicting Supreme Court Decisions Using Supervised Learning. Stanford 221.

[7] Wang, Manning. ”Baselines and Bigrams: Simple, Good Sentiment and Topic Classification.” Stanford, Department of Computer Science. (2012): Web. 8 Dec. 2017.

[8] Rudman, et al. ”The State of Authorship Attribution Studies”. Wharton School of Business. Web. 8 Dec. 2017.

[9] Joachims, Thorsten. ”A Statistical Learning Model of Text Classification for Support Vector Machines”. Web. 9 Dec. 2017.

[10] Sassano, Manabu. ”Virtual Examples for Text Classification with Support Vector Machines.” Fujitsu Laboratories Ltd. Web. 9 Dec. 2017.

[11] Hassan, et al. ”Deep learning for sentence classification.” IEEE. 5 May 2017.

[12] Ray, Laura Krugman. ”The Style of a Skeptic: The Opinions of Chief Justice Roberts.” Indiana Law Journal. vol. 83, is. 3. 2008.

[13] Ferguson, Roberts. ”The Judicial Opinion as Literary Genre.” Yale Journal of Law, vol. 2, is. 1. (2013).

C	Gamma	degrees	Kernel
0.1	0.1	2	radial
1	1	3	polynomial
5	5	4	linear
10	10	5	
100	100	6	

TABLE I. EACH COMBINATION OF THESE HYPER PARAMETERS WAS TESTED.