Judicial Authorship attribution and Influence Detection

Authors: Nicolas Muntwyler, Cyril Pomsar, Aaron Zürcher
Supervisor: Prof. Dr. Elliott Ash
University: ETH Zürich

1. Introduction

This report tries to answer two research Questions: Authorship attribution and influence detection of judicial opinion texts.We first use a similar approach as in Li, Azar et al.[1] to apply authorship attribution to opinion texts in the US. While being more accurate than the proposed method in [1] we then use the results to answer our second research question. Our goal is to find a measure for the influence of judges on eachother. First we try to find a measure for the influence of each judge on the author per opinion text. Secondly we want a measure, per curiam, for the influence of each judge on the author. Meaning we want a measure over the whole court for the influence of the judges on the author.

In the United States many opinion texts are unsigned. Especially controversial topics result in unsigned opinions. Hiding authorship removes accountability for the judges making the decision. Additionally it is harder to understand the reasoning behind a verdict. Anonymity makes it easy for a judge to just decide on an important matter without thinking twice. For these reasons alone it is obvious that determining the correct author to an opinion text is of high value. Authorship attribution is not only important in the case of unsigned opinion texts but also in various other domains like articles or government statements. In the past a big effort has been made to get authorship attribution to a high level of accuracy. Papers [1],[2] and [3] show the most common approaches, which all use natural language processing with machine learning. Most interestingly paper [1] did authorship attribution for unsigned opinion texts. We took a very similar approach and achieved an accuracy of up to 95%.

Authorship prediction:
We used an export of the lexis database [7],[8] as the corpus, which we then split into the 12 circuit courts and the supreme court. Since our dataset dates back until ca. 1800 we split each circuit court into three time periods such that we have more or less the same amount of opinion texts for each split. In order to have a more meaningful and concise dataset we then chose the 10 most frequent judges for each time period. Afterwards we cleaned the opinion texts and removed obvious author identifiers. We then encoded each opinion text as a bag of words of the most common 3000 uni-, bi- or trigrams. Based on the bag of words representation we used machine learning in order to predict the authors of the opinion texts. Our machine learning model is an SVM (Support Vector Machine) for which we achieved accuracies of up to 95%. With these high accuracies we conclude on our first research question, namely authorship prediction, with a summary of the accuracies in the

appendix [A3]. We can now use our results from the authorship prediction to answer our second question.

Influence detection:

Not only is it important to know who wrote the opinion text but also which judge had an influence on the verdict. Does a certain judge have a bigger influence than others? Is there collusion behind the scenes? Of course answering these questions is of high interest and importance. This is the goal of our second research question.

We want to use the prediction results from the author attribution as a basis. Since we get very high accuracies we can take a closer look at the prediction values. Because we are in a multiclass problem each judge gets assigned a probability of being the author. The probabilities come from how sure the model is that this judge can be the author. Therefore we name these probabilities: *confidence scores*

With the high accuracy of the authorship attribution task we also get high confidence scores. Meaning that for a correctly classified opinion we get a high confidence for the judge that wrote the text. Based on these confidence scores we can then make assumptions about the influence of the other judges on the author. In almost all cases our model predicts correctly and is very confident in its decision. You can see the mean confidence scores[1] for the first split of the first circuit court in the appendix [A2]. However in some rare cases the model is not so sure about its prediction. These are the cases that interest us the most. It is unusual for the model to have almost exclusively high confidence scores while having low confidence scores in some cases. We claim that if we get really high confidence scores we can safely assume that the writer was uninfluenced. On the other hand we believe that if our model has low confidence scores, that either more than one judge wrote that particular opinion or that the author changed his writing style. Our reasoning is that our model captures the similarity of writing styles. Each author has a distinct writing style. The model then tries to match the writing style of the opinion text with a writing style of a judge. If they are fairly similar the model will be sure in its prediction and we claim that the author was uninfluenced. However in the case where we have low confidence scores we know that the writing style of the texts don't really match the writing style of any judge. This means that the writing style of the opinion text is shifted somewhat to a mixture of the writing styles of the judes. This change of writing style could come from a prior discussion the judges had before deciding on a verdict, or if parts of the text were written by different judges. Let's assume a person A has an opinion O1 but then person B with opinion O2 comes and convinces A to also believe in his opinion O2. In this case it is common that when A writes the opinion text and has to reason why O2 is correct, he presents the same arguments B used in order to convince him. Further A will most likely use a similar choice of words as B was using when B convinced A to believe in O2. We think this originates from the fact that A will relive the discussion he

[1] We define the mean confidence score for a given author as: The mean of the confidence scores over all opinion texts written by this given author.

had with B while writing the opinion text and remembers what B said in order to convince him. Then he will write down his thoughts and the text will have similarities with the writing style of B. Therefore if we are in a case with low confidence scores we claim that either the author was influenced or that multiple judges wrote the opinion text, which in a sense is also influence. Hence we claim that with low confidence scores we have a high chance that the author was influenced. A more in depth discussion is presented in the 'Methods' part of this article.
We further show that using a bag of words approach of n-grams with a simple machine learning model can achieve high accuracy results in authorship attribution of opinion texts. While using more opinion texts we could achieve a higher average accuracy for the first split than [1] did for the same task. Additionally we decide between ten potential judges while [1] only has to decide between nine. Lastly we introduced a new measure of possible influence on the author for a specific opinion text and also per curiam.

2. Literature Review

This project consists of two parts (authorship attribution and influence detection) and since authorship attribution has already been done numerous times, we focus on influence detection as our own contribution to the field. We first present our research review for authorship attribution.

We found several papers, which are using Support Vector Machines as the used model for authorship attribution.
Juola [3] mentions how SVM's are widely used for authorship attribution and states how they generally outperform other methods like decision trees, neural networks, LDA and PCA. While he also states that in spite that SVM has usually the highest accuracies, there is no clear 'winning' model regarding performance for authorship attribution. Our decision to use SVM is also backed by Bozkurt, Bağlıoğlu and Uyar [2] who compare the performance of SVM with histogram methods, k-nearest neighborhood, Bayes classifier and k-means clustering on a dataset of 25000 Turkish newspaper articles with more than 500 articles per author. From their results one can see that SVM outperformed the other models by a long shot. (95% accuracy for SVM while Bayes classifiers are around 70% on second place)

Additionally Li, Azar et al. [1] who used authorship attribution for opinion texts of the US supreme court, which is basically what we are trying to do here, managed to get up to 83% accuracy in their test set using SVM even though they had a relatively small dataset of only around 600 opinion texts.

In order to determine which feature set is most indicative of individual writing styles we refer to Li, Azar et al. [1] who achieved good results by using a bag of words based approach on both document frequency and information gain of n-grams. They also write that using a combination of uni-, bi- and trigrams provides slightly higher accuracies than solely using unigrams. Because of their results we will also use uni-, bi- and trigrams. Usually the number of n-grams is far too big to process. Therefore one reduces the number by using a cutoff like information gain or document frequency. In paper [1] good results were achieved both with information gain and document frequency. We chose document frequency.

A good explanation for why using most frequent n-grams works so well, is given by Juola [3]. He writes that using unigrams as features is a good method to capture function words like “at” and “in” which in turn are used often and topic independently (which is very desirable in our cases since our judicial opinion texts are on numerous topics). Additionally function words have large synonymity and can be used interchangeably. An individual usually sticks to his own subset of these or uses them more frequently. So looking at function words gives cues about the authorship of a text. Adding bi- and trigrams also adds information about how these words are used in context with each other, since the meaning and therefore the usage of some function words change completely in certain contexts.

We note that most authorship attribution literature is rather old in respect to the progress of natural language processing. The newest relevant paper for authorship attribution that we found is from 2013. In the past Deep learning architectures never achieved good results in authorship attribution tasks. But with the recent progress in Deep learning with transformer based methods like BERT the accuracies have gotten much better. However we have not found any literature trying authorship attribution with these newer methods. Since we are unsure about the performance of the newer methods in respect to authorship attribution, we will stick to the older but reliable methods like SVM.

Regarding the Influence detection we could not find any related literature. In the past journalists and researchers tried to analyze judicial influence manually.

3. Data and Summary Statistics

We received our corpus from Christoph Gössmann ([7] and [8]). This corpus data is split into two parts, one [7] containing all cases with case specific information and the other one [8] containing opinion data per court case. This split was done, because one case can have several opinion texts. The opinion data and the case data are linked by a unique case identifier called dc_identifier. The first thing we do is deciding which columns might be of use later and which columns we probably won’t ever use. We decide on keeping the date, dc_identifier and dc_source column from the case data, where the dc_source column tells us, which court this case belongs to. From the opinion split we take the columns:

- dc_identifier: link between cases and opinions
- opinion_type: specifies the case type
- opinion_id: unique identifier for opinions
- opinion: the opinion text
- word_count: number of words in the opinion text
- authorship: single name signifying the author of the opinion text
- authors: list of names stating the authors of the opinion. Can differ from authorship. This column was added by Christoph Gössmann and isn’t complete.

Since we are interested in authorship attribution, we delete all rows where the authorship field is empty. We also considered removing all rows, where the authors field was empty, but comparing both resulting dataframes revealed that there are many rows with an empty authors field, but non-empty authorship field. Also no rows were found, where the authorship field was empty but the authors field filled in. So ultimately we decided on authorship, since this results in less data being ignored. We end up with 580184 rows left. Afterwards we separate the data by court. We have 13 different courts, the United States Court of Appeals of the First Circuit up to the United States of Appeals of the Twelfth Circuit and the Supreme Court of the United States. The row distribution is as follows:

- The First Court of Appeals:	24919
- The Second Court of Appeals:	49365
- The Third Court of Appeals:	52105
- The Fourth Court of Appeals:	38111
- The Fifth Court of Appeals:	65027
- The Sixth Court of Appeals:	55107
- The Seventh Court of Appeals:	45052
- The Eighth Court of Appeals:	52992
- The Ninth Court of Appeals:	65965
- The Tenth Court of Appeals:	53737
- The Eleventh Court of Appeals:	24796
- The Twelfth Court of Appeals:	18839
- The Supreme Court:	34169

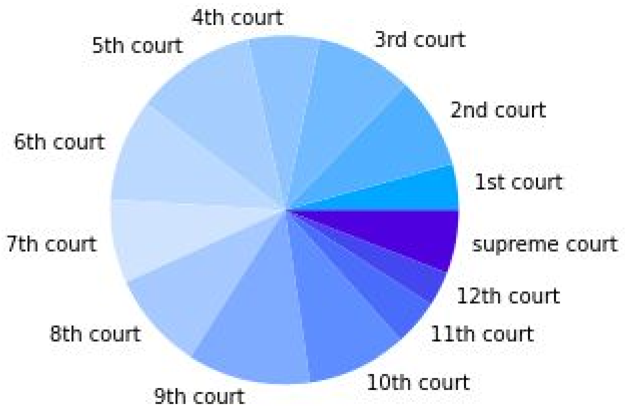


Figure 1: Distribution of opinion texts among the courts

Then, for every court apart from the first, we search the names of the 30 most frequent authors in the authorship column. For the first court we handpick the names by using the wikipedia article about the first court[9]. One of the most frequent authors is ‘PER CURIAM’. So we obviously have to remove those authors from our lists. Next we remove all data, where the author is not one of the 30 most frequent authors. During this process we also remove all rows, where the word count of the opinion text is less than 50. Those opinions with less than 50 words are most often just “AFFIRMED”, or a short sentence containing the word affirmed. We are now left with

- The First Court of Appeals:	19833
- The Second Court of Appeals:	27453
- The Third Court of Appeals:	23754
- The Fourth Court of Appeals:	16480
- The Fifth Court of Appeals:	26673
- The Sixth Court of Appeals:	22944
- The Seventh Court of Appeals:	31407
- The Eighth Court of Appeals:	28778
- The Ninth Court of Appeals:	20075
- The Tenth Court of Appeals:	25971
- The Eleventh Court of Appeals:	10805
- The Twelfth Court of Appeals:	12214
- The Supreme Court:	18629

rows per court. This leaves us with a total of 285016 signed opinion texts.

The following actions are done for each court separately.

Now we begin to clean the opinion texts. First we try to remove all words indicating the title of the author(s) like “Senior Circuit Judges” or “CIRCUIT JUSTICE” and all alterations, which there are about 25 of them. Next we remove the names of all authors from all opinion texts. Since the names are sometimes written differently in the opinion text, we try to remove all alterations of the given last names. Some authors also add their first name in their opinion text. To find the alterations of the family names and the authors who add their first name to the text, we take a sample of ten texts of a given author. And then we manually look if they mention their first name in the texts and in what form their family name is written in those texts. If they do mention their first name, we remove them from our corpus together with all found alterations of the family name. Then we remove all numbers from the texts. Now we are left with artefacts from cleaning like “..,” and “§”. We try to remove all of them individually.

Finally we decide to split up each court but the last into three parts. We calculate at which year we have to split for each court separately, so that the splits are roughly evenly sized. Then, for each split individually, we take the 10 most frequent authors and only keep the opinion texts written by those 10 authors. Because the three splits in the supreme court were too small we only split it into two parts. Here is some data about the resulting corpus:

Court:	Years for Split 1:	Years for Split 2:	Years for Split 3	Split 1 size:	Split 2 size:	Split 3 size:	Total:
First	<1980	1980-2000	>2000	5627	5363	5584	16682
Second	<1922	1922-1961	>1961	8462	8054	8454	24989
Third	<1983	1983-2007	>2007	5802	5362	6200	17364
Fourth	<1968	1968-1991	>1991	4656	4603	4700	13959
Fifth	<1958	1958-1985	>1985	7756	7367	7742	22865
Sixth	<1993	1993-2007	>2007	5943	5151	6424	17724
Seventh	<1979	1979-2003	>2003	6876	11483	7278	20587
Eight	<1983	1983-2000	>2000	7717	7221	7595	22533
Ninth	<1941	1941-1981	>1982	5783	5500	5670	16905
Tenth	<1982	1982-2000	>2000	6214	5836	6509	18559
Eleventh	<1990	1990-2002	>2002	2672	2379	2656	7707
Twelfth	<1998	1998-2007	>2007	3024	2878	3031	8933
Supreme	<1956	>1955	-	6829	6583	-	13488

Figure 2: Size distribution of the splits and their respective date range

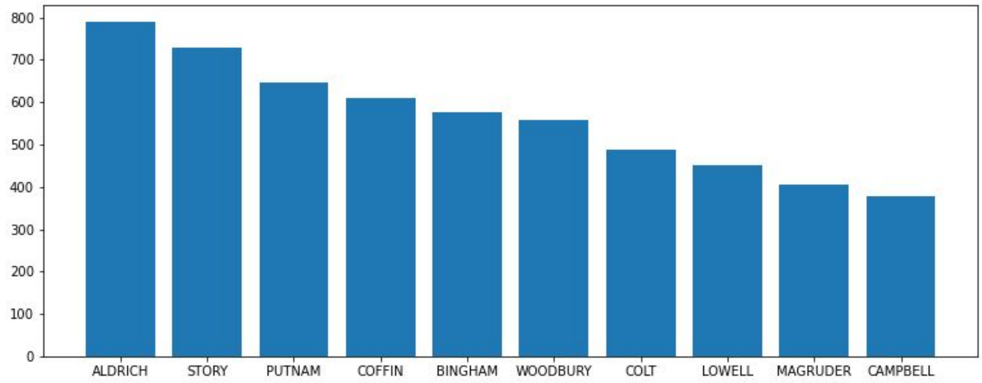


Figure 3: top 10 most frequent authors in the first split of the first court

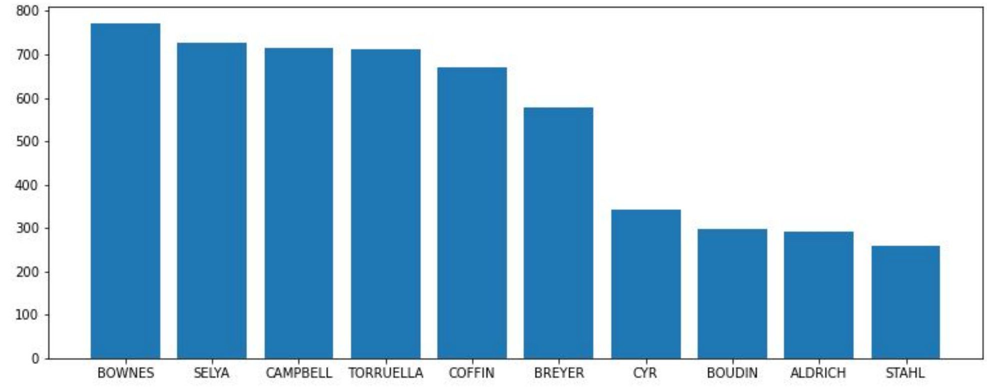


Figure 4: top 10 most frequent authors in the second split of the first court

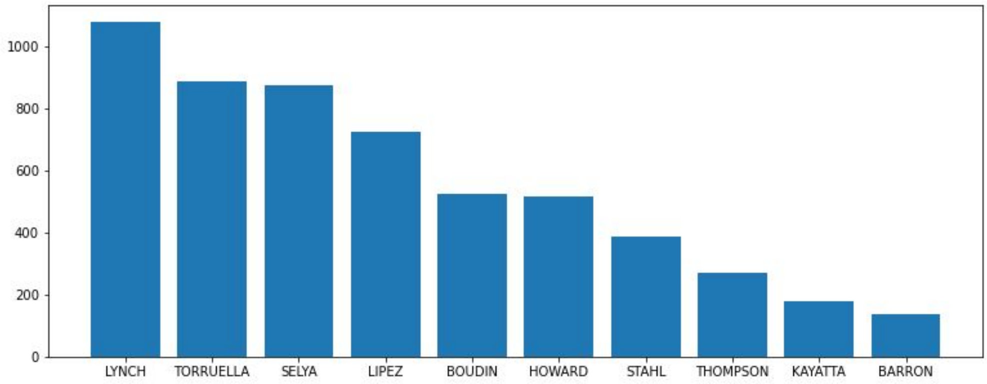


Figure 5: top 10 most frequent authors in the third split of the first court

4. Methods

We remember that we are trying to answer two research questions. First we conduct authorship attribution and secondly use the results from the authorship attribution task to make statements about the influence judges had on eachother.

4.1 Authorship Attribution:

Our first task is to achieve an authorship attribution prediction with an as high as possible accuracy. Our approach is to use supervised machine learning in order to identify characteristics (features) for each judge's writing style. The used features come from the opinion texts and the corresponding label will be the judge, who wrote this opinion text. In previous papers ([1],[2],[3]) high accuracy authorship attribution tasks were achieved with such a supervised machine learning approach where a bag of words of uni-,bi- and trigrams, was used to encode the opinion texts as a feature set. Additionally in paper [4], document frequency as a method for dimension reduction had better results than using other approaches like Information-gain. Therefore we chose to generate uni-,bi- and trigrams for each cleaned opinion text and then chose the 3000 most frequent ones. We also tried using a function word feature set since in paper [2] they achieved high accuracies by only using a bag of words approach with function words. However our accuracies with this feature set were below the accuracies we could achieve with the n-grams. For example in the first split of the first circuit court we have an accuracy of 0.88 with the n-gram feature set but only 0.67 with the function words feature set. Therefore we decided to use n-grams as our main feature set. We stored these n-grams as a bag of words encoding for each opinion text. This means that our dataset has 3000 columns, one for each chosen n-gram. The number of rows equals the number of opinion texts. An entry in this matrix tells us how many times the n-gram was found in the corresponding opinion text.

Afterwards we standardized each feature. Hence for each column we subtracted the mean of that column and divided it by its standard deviation. Standardizing our data resulted in a better accuracy. After the generation of our feature set and labels we divided our data into training (80%) and test (20%) data. We use the training data solely to train our model. We used cross-validation to get accurate accuracy results and do model evaluation. After some testing we concluded that SVM with the polynomial kernel and a degree of one gave the best results. Other considered models were:

- SVM with an rbf kernel
- SVM with poly kernel and 2 degree
- Random Forest

After we used the training data to train our model, we used the model to predict our test data. The accuracies for each circuit court and for each time split are listed in the Appendix [A3]. With these accuracies we conclude the authorship prediction task.

4.2 Influence Detection:

We now want to use these results in order to make statements about the influence of judges. Our goal is to find a measure, per opinion text, for the influence of each judge on the author. This means we want to know that if a certain judge A had a noticeable influence on the decision judge B made. Not only do we want to answer this question per opinion text but also over all cases.

We define the influence of a person A onto a person B as the influence A had on a text written by B.

Therefore our goal would be equivalent to: Find a measure, per opinion text, for the influence of each judge on the author's text. So how can we achieve this? Since we basically only have the opinion texts as a source of input, we make the following assumption:

Assumption 1:
If a text is influenced by a person A, then the text's writing style will show similarities to the writing style of A.

This means that if a person A is influenced by a person B, then a text written by A will have a more similar writing style to B's writing style, than a text with A being uninfluenced. Now in order to capture how similar writing styles are, we need to have a measure for the similarity of writing styles.

How to capture a writing style:
Each judge has its own writing style. We create a hyperspace of all the different writing styles. So the writing style of judge A would be a subspace in the hyperspace of writing styles. So if person A writes a text, we capture this text's writing style as a point in our hyperspace. Since A is the author, we expect that this point then lies inside As subspace. But how can we create such a hyperspace?

From hyperspace to featurespace:
Each writer has his choice of words, average word length, punctuation style etc. We capture some of these attributes as features. So the hyperspace of these features, which we call featurespace, represents the hyperspace of writing styles (Assumption 2).

Assumption 2:
Let subspace(x) be the subspace of the writing style of the person x.
A text t is in the subspace(x) of the hyperspace if and only if t is in the subspace(x) in our featurespace.

This assumption also implies that the closer a text's writing style is to a writing style of a person x, the closer it is to the subspace(x). Although we can never find a completely correct featurespace, we can try to find a good approximation. In order to find a good approximation we want to use machine learning. Our model should learn the different subspaces of the different authors. This is exactly what we did on our first task. By doing authorship attribution we implicitly learned the different subspaces for the writing styles of the judges. And with the high accuracies we can see that we have found a good approximation for the ground truth hyperspace for writing styles.

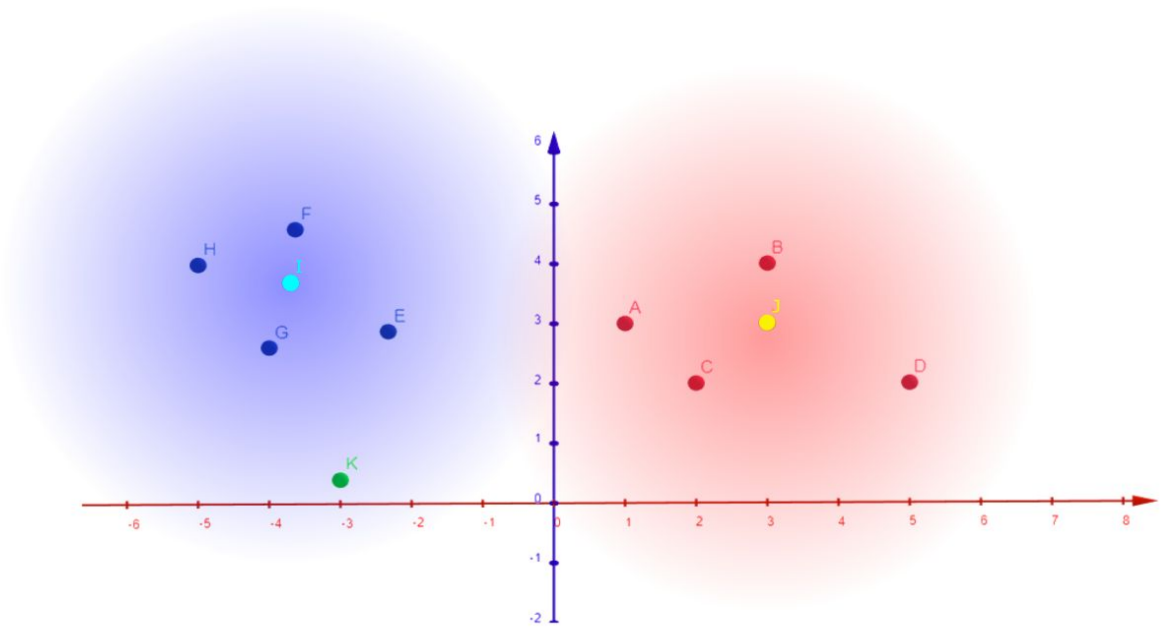


Figure 6: This Figure represents a visualization of the hyperspace of writing styles. In this example we only have two authors Blue and Red. The hyperspace is represented by only two features (x-Axis and y-Axis). In practice we would have thousands. The blue points (E,F,G,H) are opinion texts of Blue represented in the hyperspace. The same holds for the red points (A,B,C,D) for Red. The cyan point I stands for the average writing style of the blue Author. It is the midpoint of the subspace of Blue's writing style, which is represented by the faded blue ellipse. The closer we are to I the more the text has the writing style of Blue. (In practice we will never know I). The same holds for Red with midpoint J.
Our model would predict that the green point K (representation of an opinion text in our hyperspace) would be written by Blue, since it is closer to I than to J.

We now have found a features space that approximates the hyperspace of writing styles. Remember that our goal was to capture how similar the writing style of a text T is to each judge's writing style. So we want a measure of how close the writing style of T is to each author's subspace. In our authorship attribution we use SVM with a polynomial kernel with degree one. We then can use the built in 'predict_proba' function which gives us a probability distribution for each judge's likelihood to be the author. This means that we are for example to 0.9 sure that it was judge B and 0.1 sure that it was judge A (in the case of only two judges). In the paper [6] about the implementation of the 'predict_proba' function one can read that these probabilities are based on the distance of the learned subspaces of the judges to the text's position in the learned featurespace for the writing styles. Therefore the result of the 'predict_proba' function directly represents the distance to each judge's subspace. This means the writing style similarity is directly represented by the probabilities of the 'predict_proba' function. For example our 'predict_proba' function could return such an output:

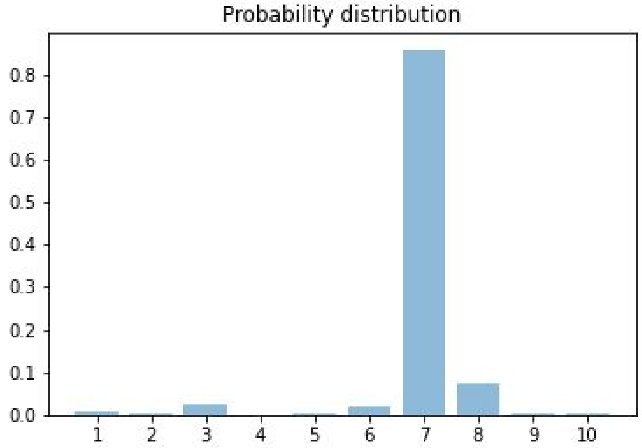


Figure 7: Example probability distribution with 10 judges

In this case we can see that our model is 0.8 sure that judge 7 wrote this opinion text. While for other judges we have very low certainties of about 0.05. Therefore the similarity between the writing style of judge 3 and the writing style of this opinion text is much higher than the similarity of the text and another author.

So after all that we have a measure of similarities for writing styles. Unfortunately similar writing styles do not imply influence. Meaning, if a text has similarities with the writing styles of two persons we have two possibilities:

(1) Both had an influence on the text

(2) Both persons have similar writing styles (and only one or both had influence)

Note that Assumption 1 still holds, it's just the reverse direction that does not hold.

Unfortunately it is impossible to differentiate between those two possibilities. So in order to have an intuition on which of these cases hold we look at the overall mean confidence scores. For all texts written by judge A, we took the prediction values of the 'predict_proba' function and then took the mean for each judge separately. In other words for each judge we get a score of how sure the model is on average that this judge was the author for all cases that author A wrote. This means that we expect that the confidence score will be highest for judge A, since we have a high prediction accuracy and our prediction is based on just taking the author who got the highest confidence score from our model for that case. Let's make an example:

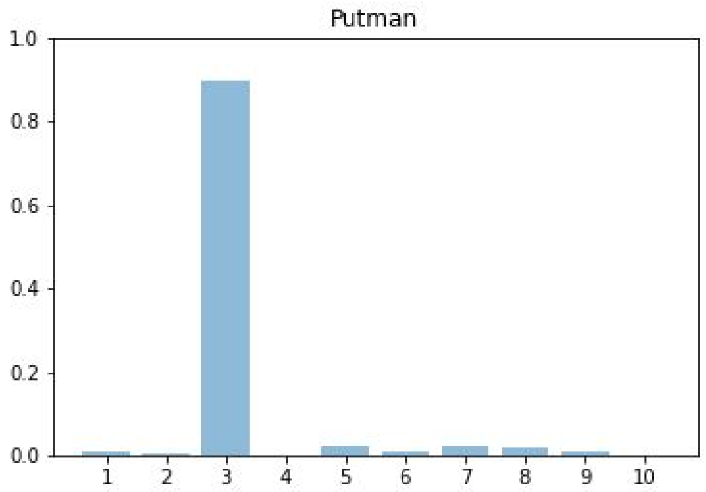


Figure 8: Overall Mean Confidence Scores of 3rd judge (Putman) in First circuit Court Split 1

In figure 8 we can see the overall mean of the confidence score values for each judge for all cases that judge 3 wrote. We can see that we have a very high mean confidence score for judge 3. In this case we even got for some opinion texts a confidence score of over 99.9%. Taking this overall mean of confidence score values gives us an intuition on which of the two possibilities (1) and (2) holds. For this specific case we argue that our model could reliably differentiate between similar writing styles. We reason this way because the mean confidence scores of the judges that did not write the opinion texts were really low, while in contrast the mean confidence score for the judge that wrote the opinion texts, was extremely high. If the author judge had a similar writing style as another judge and our model could not reliably differentiate these two writing styles, the model would be much more unsure about the cases. Hence the confidence scores would be significantly higher for judges with similar writings styles. Therefore if we now would look at a specific case for which the confidence values would have a spike at two different judges A and B, where A was the author, we claim that author B had an influence on judge A. However in another case however we can see a different result:

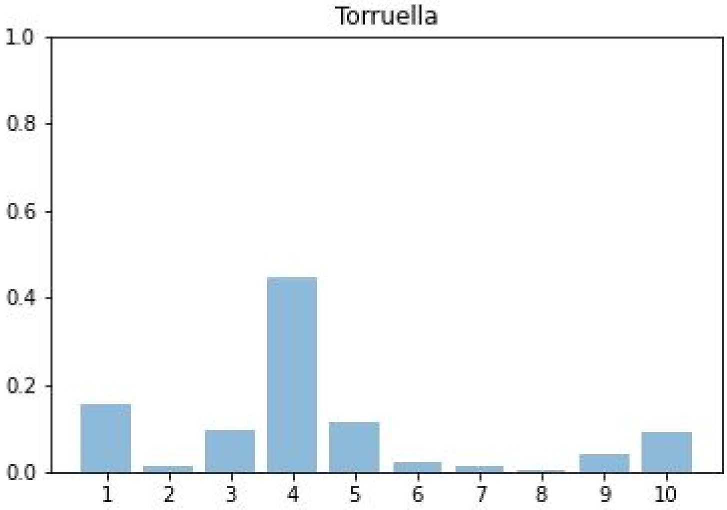


Figure 9: Overall Mean Confidence Scores of 3rd judge (Torruella) in First circuit Court Split 2

For this example we can see that the overall mean of the confidence score for each judge for all cases that the 3rd judge (Torruella) wrote, have a much more even distribution than in the previous case. The mean confidence scores for authors that did not write the opinion text are much higher than before, and consequently the mean confidence score of the judge that was the author of all these option texts is much lower. If we now go back to the two possibilities (1) and (2) we can state for this case that either:

1. In almost all cases the other judges had an influence on the author.
2. The writing styles of the judges were much more similar than in the previous case and the model had more trouble to distinguish different writing styles.

In the case for (1) we are finished and can analyse our result with this assumption in mind. However in the case that (2) holds, we would like to subtract the overall mean confidence score for each judge that did not write the opinion text, when we make a new prediction.

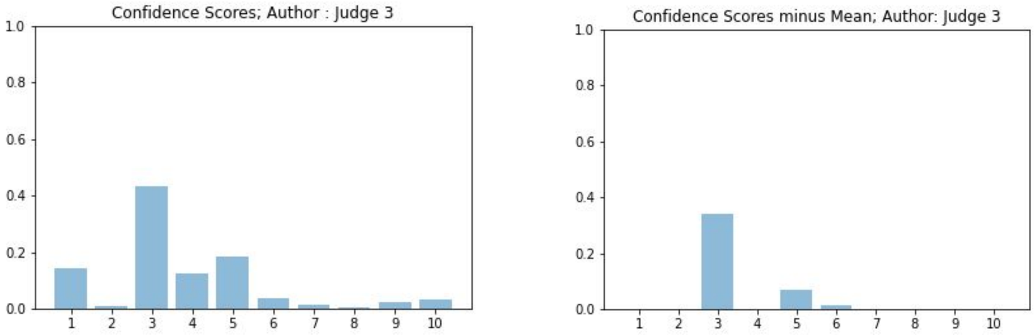


Figure 10: Case 26, First Circuit court, Split 2

We argue that this difference will be the influence that person had on the writer. The reasoning is that our model simply captures the similarity of writing styles. Now as we mentioned earlier a text written by judge A can have a similar writing style as author B would write. This similarity can come from two factors. The first being that B just has a similar writing style. The second being that B influenced A. Therefore if we now assume that the author was mostly uninfluenced the overall mean of the confidence score values should represent how similar the writing styles would be. Hence if we now subtract this overall mean from each judge we implicitly take away the factor of having a similar writing style. This leaves us with just the first factor, which represents the influence. In practise just subtracting the overall mean for the judges that did not write the opinion text does not make much sense, since some values could fall below 0, which would make no sense. Therefore if the value would fall below 0 we would just set it to 0. With these new values we would then build a distribution function again.

Per Curiam analysis

So far we analyzed the influence for a specific opinion text. However we already introduced the overall mean confidence scores, which we can use to analyze the influence per curiam. If the mean confidence scores only have one spike like in figure 8, we can safely state that this judge was per curiam uninfluenced. However if our graph looks more like figure 9 we either have that the judge was influenced, seen over all cases, or that he has a similar writing style to the other judges and our model has problems differentiating them. Which case holds can't be proven. But if we assume that our model can distinguish similar writing styles accurately we could state that this author was influenced.

Summary

We first solve the task of authorship attribution by using a bag-of-words approach and use an SVM predictor. Afterwards we use the results of our authorship prediction for our influence detection. For the per opinion text analysis we calculate the confidence scores of our predictor for the opinion text in question. Since we can't distinguish between influence and similar writing styles we use the overall mean of the confidence scores to make reasonable assumptions. We differentiate between three cases:

1.) Overall mean is low for all non-author judges.
2.) Overall mean is more evenly distributed
    a.) We claim this is because the author was influenced in almost all cases.
    b.) We claim this is due to similar writing styles.

In the cases 1.) and 2.a) we can directly take the result of our predictor and make statements about the influence. Only for case 2b) one would have to subtract the overall mean before making statements about the influence.

If we want to do a per curiam analysis we would only look at the mean confidence scores. If we are in case 1 we can assume our author was uninfluenced. However for case 2 we either conclude that this author was influenced in many cases or we would assume that he has a similar writing style as the other judges.

5. Results

In the following section we present our results. We first state the overall results and then show four short case studies.

In the authorship attribution task we achieved the accuracies presented in the appendix [A1]. The highest accuracy was for the first split in the fourth circuit court of 95%. Random guessing would result in just 10%. For the first split, we achieved an average accuracy of 85% whilst for the second and third split an accuracy of 73%. Our main contribution however is the prediction of influence. Our goal is to return a measure of influence, i.e. how much the author was influenced by which judge. As we explained in section 4 we have to differentiate between different cases before we make statements about the influence.

In order to get the prediction of influence for a given opinion text one can follow the algorithm below as a Guideline.

Algorithm to output our influence measures:
Given: An opinion text with its probability values of our prediction and the mean confidence scores of its author:

1. *Look at the overall mean confidence score of the given author. If the maximum overall mean score of the non-author judges has a value above 0.2 or if the overall mean score of the author judge is not below 0.5 we decide to go to 2. Otherwise we go to 3.*

2. *Return the probability values of our prediction as the influence scores. END.*

3. *We return two sets of influence scores. One (a.) assuming that the author's writing style is different and one (b.) assuming that the writing style is similar to all other authors.*

    a. *Return the probability values of our prediction as the influence scores.*

    b. *We subtract from all the probability values the mean confidence scores. Any value below zero will be instead changed to zero. Return this result as the influence scores. END*

In order to give some examples we conducted 4 case studies, each covering a different case with a different outcome of influence.

Case Study 1 **(clear mean, one spike)**

We choose the 23rd case of the first circuit court of the first split. The correct author for this opinion text is judge 7. He has the following overall mean confidence scores:

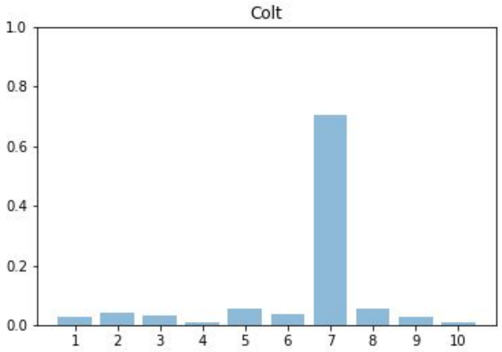


Figure 11: Mean Probability Distribution of the 7th judge

We can see that our model is usually very sure about its prediction for texts written by author 7. Now for the 23rd opinion text our model has the following prediction output:

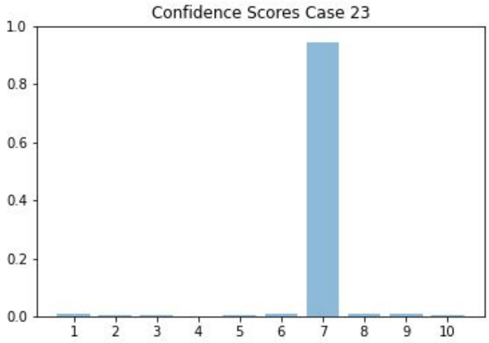


Figure 12: Probability Distribution of the 23rd opinion text

We observe that our model is absolutely sure that it was judge 7, which is indeed the correct author. Since our model is so sure that it predicted the correct author we can assume that the author was uninfluenced by the other authors when writing the opinion texts. The measure of how much which judge influenced the author are exactly the probabilities in the Figure.

Case Study 2 **(clear mean, two spikes)**

We choose the 102nd case of the first circuit court of the first split. The correct author for this opinion text is judge 6. This judge has the following overall mean confidence scores:

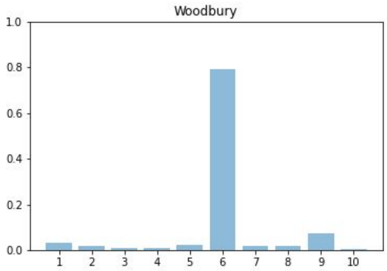


Figure 13: Mean Probability Distribution of the 6th judge

We can see that our model is usually very sure about its prediction for texts written by author 6. Now for the 102nd opinion text our model has the following prediction output:

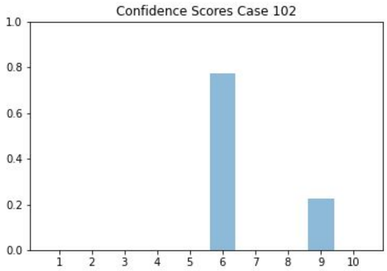


Figure 14: Probability Distribution of the 102nd opinion text

The correct author for this opinion text is author 6. However in this case we can see that our model is not 100% sure about its decision. For some reason it also thinks that judge 9 could be the correct author. We conclude that judge 9 had influence on judge 6.

Case Study 3 **(not clear overall mean, spike from unsimilar writing style)**
We choose the 896th case in the first split of the first circuit court. The correct author for this opinion text is the 10th judge. He has the following overall mean confidence scores:

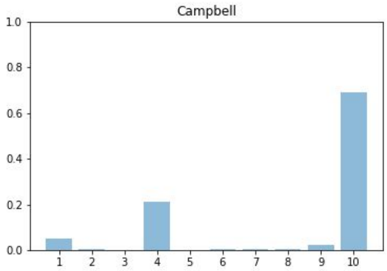


Figure 15: Mean Probability Distribution of the 10th judge

We observe that for this judge the model is in average not as sure as in the previous cases for opinion texts written by author 10, since the overall mean score for the 4th author is over 0.2. This means that in many cases the model is contemplating between judge 10 and judge 4 to be the correct author in cases where judge 10 would be the correct choice. In this case we can now either have that judge 4 had an influence on judge 10 in many cases or that judge 4 has a similar writing style.

If we believe in the first case we are now finished and can just return the returned probability distribution for the 896th case as our influence scores, which is:

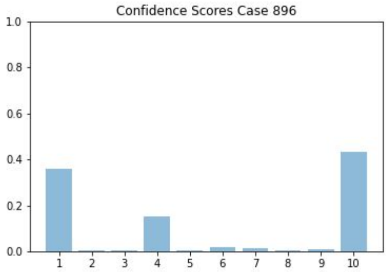


Figure 16: Probability Distribution of the 896th opinion text

We state that for this case we believe that judge 4 had influence and that judge 1 had a very strong influence on the opinion text written by judge 10.

However if we now think that judge 4 just has a similar writing style as judge 10, we can subtract the overall mean from the confidence scores of the 896th opinion text and get:

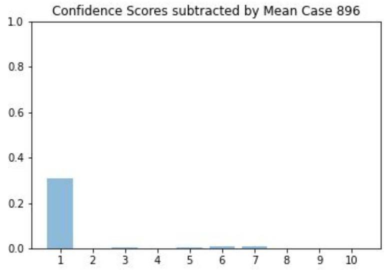


Figure 17: Probability Distribution of the 896th opinion text subtracted by the Mean

Here we see that the 4th judge has no influence anymore, but the value for the influence score of the 1st judge has hardly changed. In both assumptions we see that judge 1 has a similar influence on the text. So even though we are unsure about the influence the 4th judge had on the text, we can confidently say that the 1st Judge strongly contributed to the creation of the 896th opinion text.

Case Study 4 **(not clear overall mean, spike from similar writing style)**
We choose the 185th case in the first split of the first circuit court. The correct author for this opinion text is the 10th judge. He has the following overall mean confidence scores:

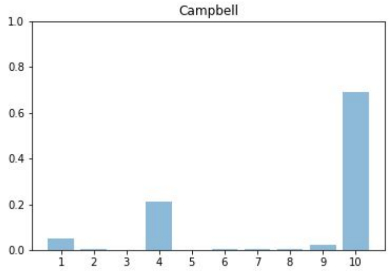


Figure 18: Mean Probability Distribution of the 10th Judge

Since it's the same judge as in the 3rd case study we also observe the 4th judge either has a similar writing style or is influencing the 10th judge. So we are in part 3 of our algorithm.

If we proceed to step a) and believe that the 4th judge is influencing our author, we would output the following Influence Scores:

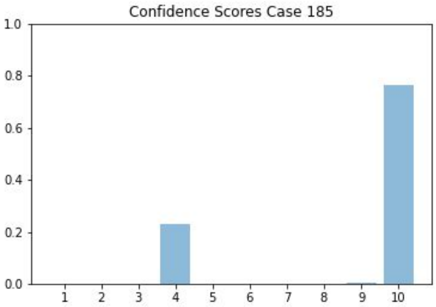


Figure 19: Probability Distribution of the 185th opinion text

However, if we believe that they have similar writing styles, we would go to step b) and the resulting influence scores are:

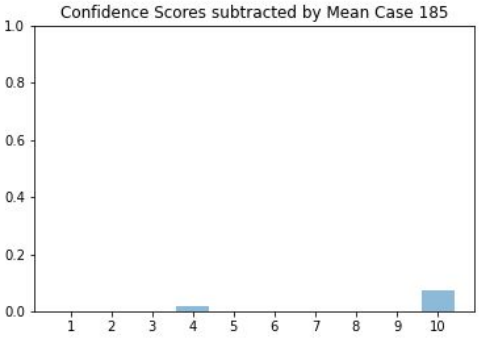


Figure 20: Probability Distribution of the 185th opinion text subtracted by the Mean

As you can see, the resulting influence scores differ greatly for the 4th judge. The first case would say that the 4th judge had quite an influence on the writing of the text. In contrast the second case would suggest that the author was mostly uninfluenced. Like previously mentioned we can’t know, which of the two cases are true.

Per Curiam Influence analysis:
As we have previously stated our measure of influence is unable to differentiate between similar writing styles and “being influenced”, since we base our measurements on writing styles. That being said, if we assume that our model has no problems deciding between writing styles, then the mean confidence scores would represent exactly the per curiam measurement of influence for each judge on the author. These confidence scores for the first split of the first circuit court can be seen in the Appendix [A2].

For the first split in the first Circuit court we can see that the judges Aldrich, Story, Putman, Bingham, Woodbury, Colt and Margruder were probably uninfluenced.
However one can observe that judge Campbell could have been influenced by judge Coffin since we have a small spike for judge Coffin in the overall mean confidence scores of judge Campbell. Similarly Coffin could also have been influenced by judge Campbell.
Lastly judge Lowell has very low mean confidence scores in other judges distributions and in his own graph he only spikes to 0.6, while other judges have mean scores of over 0.8 in their respective graph. We can assume that judge Lowell was not a very dominant person and he probably did not influence any other judge. On the other hand he could have been slightly influenced by all the other judges.

6. Conclusion

In this article we presented an approach for authorship attribution. We then used the results of the authorship attribution to find a measure of influence. This is the first try on detecting influence and could be used as a first step to find cases which would be interesting to investigate further. In order for our method to work reliably one would have to score high accuracies in the authorship attribution task. This does not hold for all splits and courts that we covered. Future work could improve the accuracy and therefore produce more reliable influence scores. The presented method for influence detection is also limited by the fact that it can’t distinguish if two judges just have a similar writing style or always influence each other in a specific opinion text.
We believe that this problem cannot be solved with our assumptions.

References:

[1] Using algorithmic attribution techniques to determine authorship in unsigned judicial opinions (2013)

William Li, Pablo Azar, David Larochelle, Phil Hill, James Cox, Robert C. Berwick, Andrew W. Lo

http://people.csail.mit.edu/wli/papers/algorithmicattribution.pdf

[2] Authorship Attribution: Performance of various features and classification methods (2007)

Ilker Nadi Bozkurt, Özgür Bağlıoğlu, Erkan Uyar

https://www.researchgate.net/publication/4321080_Authorship_attribution

[3] Authorship Attribution (2008)

Patrick Juola

https://www.researchgate.net/publication/308073726_Authorship_attribution

[4] Feature Selections for Authorship Attribution (2000)

Jacques Savoy

https://dl.acm.org/doi/pdf/10.1145/2480362.2480541

[5] Measuring Polarization in Westminster Systems (2018)

Peterson and Spirling

Classification Accuracy as a Substantive Quantity of Interest: Measuring Polarization in Westminster Systems

[6] SVM-proba function https://www.csie.ntu.edu.tw/~cjlin/papers/svmprob/svmprob.pdf

Received Corpus from Christoph Gössmann:

- **[7]** cases: https://www.dropbox.com/s/84gw1qzh299pc5j/20200616_lexis_cases_circuit.csv.gz?dl=0
- **[8]** opinions: https://www.dropbox.com/s/mgnllaal5583sos/20200616_lexis_opinions_circuit.csv.gz?dl=0

[9] Wikipedia article about the Court of Appeals of the first Circuit:

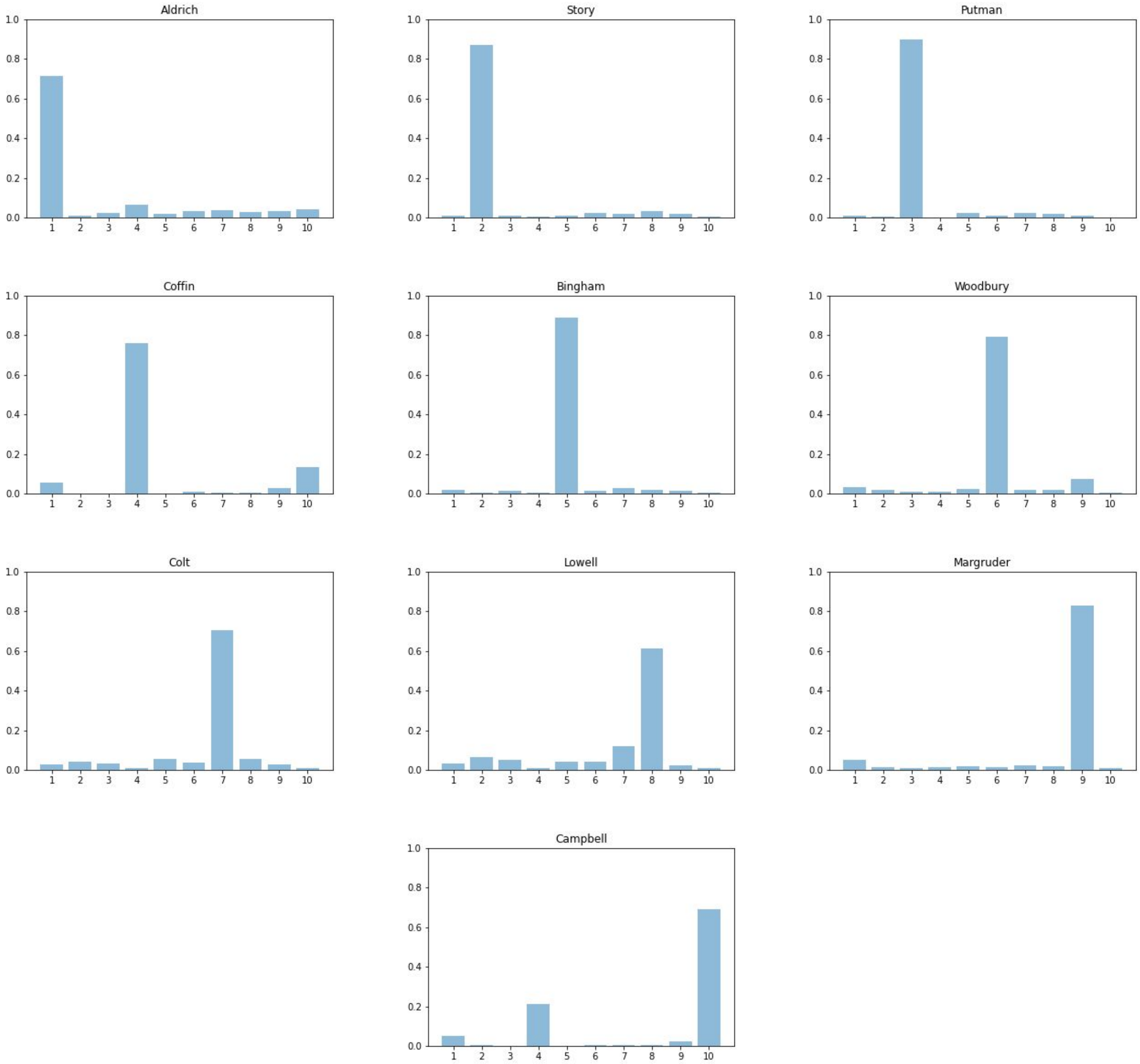
https://en.wikipedia.org/wiki/United_States_Court_of_Appeals_for_the_First_Circuit

Appendix:

[A1] United States Court of Appeals of the First Circuit

Split 1:		Split 2:		Split 3:	
Judge name:	text count:	Judge name:	text count:	Judge name :	text count:
Aldrich	789	Bownes	771	Lynch	1079
Story	727	Selya	727	Torruella	887
Putnam	645	Campbell	714	Selya	874
Coffin	611	Torruella	712	Lipez	724
Bingham	576	Coffin	671	Boudin	525
Woodbury	557	Breyer	577	Howard	516
Colt	488	Cyr	343	Stahl	389
Lowell	450	Boudin	298	Thompson	271
Magruder	405	Aldrich	291	Kayatta	180
Campbell	379	Stahl	259	Barron	139

[A2] Circuit Court 1, Split 1, Mean Confidence Scores per author



[A3] Accuracies of each Circuit court and split

Accuracies	Split 1	Split 2	Split 3
Court 1	0.88368	0.77366	0.76555
Court 2	0.85879	0.88363	0.81478
Court 3	0.85038	0.70163	0.70565
Court 4	0.95436	0.74301	0.76809
Court 5	0.93924	0.69002	0.68348
Court 6	0.87319	0.64709	0.77101
Court 7	0.89000	0.74995	0.82146
Court 8	0.89960	0.72336	0.84391
Court 9	0.88423	0.81250	0.54648
Court 10	0.87739	0.61259	0.74153
Court 11	0.56822	0.64567	0.69459
Court 12	0.69917	0.74707	0.60990
Supreme	0.92553	0.78501	-
Average	0.854	0.732	0.731