# Affirm or Reverse? Using Machine Learning To Help Judges Write Opinions

Sharan Agrawal, Elliott Ash, Daniel Chen, Simranjyot Singh Gill, Amanpreet Singh, and Karthik Venkatesan

June 29, 2017

**Abstract**

The U.S. federal court system relies on a system of appellate review where higher courts can either affirm or reverse the decision of the lower courts. We build a machine learning model to predict these appellate affirm/reverse decisions using the text features of the lower-court case under review. The data include all Supreme Court and circuit court decisions since 1880, and most district court decisions since 1923. We use a wide of classification techniques based on n-grams and convolutional neural networks. We achieve an accuracy of 79% in predicting the affirm/reverse decision of circuit courts using district court case text. We achieve 68% accuracy in predicting Supreme Court decisions using circuit court case text.

## 1 Introduction

In the United States, the federal court system has three main levels: district courts (the trial court), circuit courts which are the first level of appeal, and the Supreme Court of the United States (SCOTUS), the final level of appeal in the federal system. The district courts are the general trial courts of the federal court system. All cases are heard in district court and a verdict is announced. These district cases sometimes are appealed in the circuit court where a decision is made to affirm or reverse the decision of district court partly or completely. After circuit court, the final appeal can be made in Supreme Court of United States which decides to affirm or reverse the decision of the circuit court.

This paper builds on developments in machine learning and the prior work of **(author?)** [10], **(author?)** [12], **(author?)** [4] and **(author?)** [7]. Previously, there has been work done on predicting circuit court reversals in supreme court by Katz et al using randomized tree method where they achieved an accuracy of 69.7% **(author?)** [6]. Martin et al used a classification tree with six features to classify cases as affirmed or reversed based on binary answers to those features. They compared their cases with actual legal experts' responses for a particular case and found out that model was 16% more accurate than the experts themselves. There have also been quantitative studies dated sixty years back for predicting supreme court behavior **(author?)** [9].

First, we have built a classifier model for predicting district court case reversal in circuit court. We delineate what plays as an important factor when a court decision is made. We use the text of district court cases to build a prediction model based on ngrams and word embeddings.

Second, we trained classifier models to predict decisions of SCOTUS. We delineate what plays as an important factor when a court decision is made.

# 2 Data

## 2.1 Judge Biographies

We obtained the set of judge biographical characteristics from the Appeals Court Attribute Data,[1] Federal Judicial Center, and previous data collection.[2] From these data we constructed dummy indicators for whether the judge was female, non-white, black, Jewish, catholic, protestant, evangelical, mainline, non-religiously affiliated, whether the judge obtained a BA from within the state, attended a public university for college, had a graduate law degree (LLM or SJD), had any prior government experience, was a former magistrate judge, former bankruptcy judge, former law professor, former deputy or assistant district/county/city attorney, former Assistant U.S. Attorney, former U.S. Attorney, former Attorney-General, former Solicitor-General, former state high court judge, former state lower court judge, formerly in the state house, formerly in state senate, formerly in the U.S. House of Representatives, formerly a U.S. Senator, formerly in private practice, former mayor, former local/municipal court judge, formerly worked in

[1] http://www.cas.sc.edu/poli/juri/attributes.html

[2] Missing data was filled in by searching transcripts of Congressional confirmation hearings and other official or news publications on Lexis (**?** ).

the Solicitor-General's office, former governor, former District/County/City Attorney, former Congressional counsel, formerly in city council, born in the 1910s, 1920s, 1930s, 1940s, or 1950s, whether government (Congress and president) was unified or divided at the time of appointment, and whether judge and appointing president were of the same or different political parties.

## 2.2 Case Data

The district court opinions were obtained from CourtListener (courtlistener.com). This includes the raw text and metadata of approximately 280,000 District Court opinions from 1924 to 2013. This dataset, even though incomplete, represents a large set of decisions of the district courts during this time frame.

The circuit court opinions were obtained from Bloomberg Law (bloomberglaw.com). Circuit court data contains text and metadata for 387,000 cases for the years 1880 through 2013.

The Supreme Court data was obtained from the Supreme Court Database web site. This comprises the approximately 4300 circuit court cases that were reviewed by the Supreme Court.

## 2.3 Linking Upper Court to Lower Court

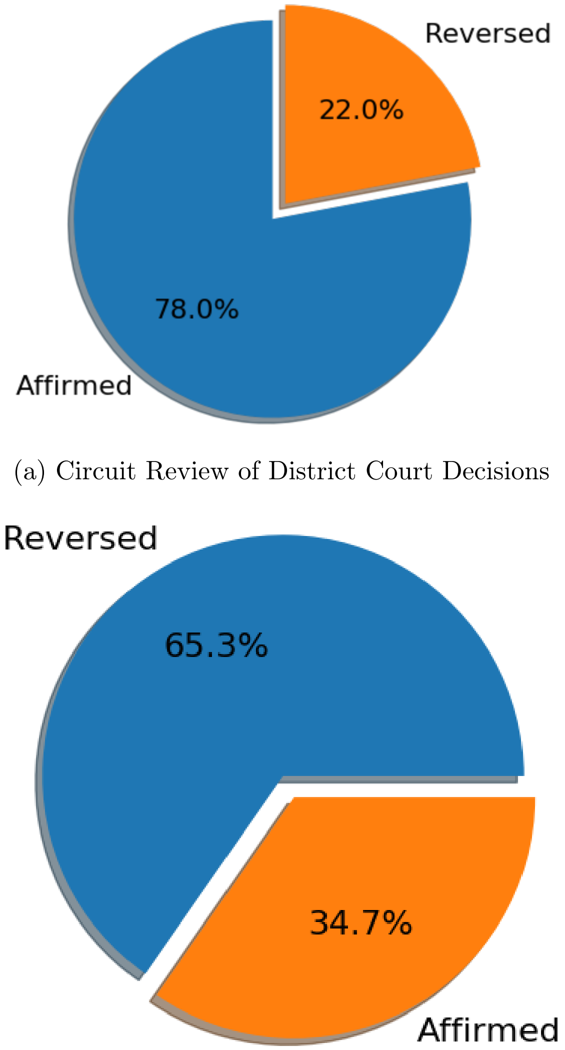
An important contribution of this project is to link upper court to lower court cases for joint analysis. The link from circuit to supreme was achieved using the case identifiers in the Supreme Court Database.

We used case name and metadata to link the district court case with the corresponding circuit court case. We considered a similarity score of 0.97 to match the case name with filters of metadata containing court dates and circuit numbers. Out of the 280,000 district cases, we found around 40,000 matches to the corresponding circuit cases. However, 25,000 of these were unpublished and we had no information if they were affirmed or reversed. We didn't took the case that partly affirmed or reversed in our dataset. Our final dataset consists of 15,000 cases with affirm/reverse information.

## 2.4 Target Variable: Affirm/Reverse

The key target variable is whether the upper court (circuit or supreme) affirmed or reversed the decision of the lower court. As initial summary statistics, Figure 1 reports

Figure 1: Affirm/Reverse Proportion



(a) Circuit Review of District Court Decisions

(b) Supreme Court Review of Circuit Court Decisions the averages for our merged data set. Circuit courts affirm district courts 78 percent of the time. In contrast, the Supreme Court affirms circuit courts 34.7 percent of the time.

Note that this difference is due in part to discretionary review at the Supreme Court level. Circuit courts are required to review appeals from district courts. But the Supreme Court can reject appeals and not review them. If these rejections were coded as affirm, there would be a higher proportion affirm that would be closer to what we see with circuit-to-district.

These proportions are important to bear in mind when evaluating the output of the prediction models described below. A naive prediction that guessed “affirm” would predict district-to-circuit review with 78 percent accuracy. A naive prediction that guessed “reverse” would predict circuit-to-supreme review with 65 percent accuracy.

As the data is imbalanced in nature, a baseline model which predicts all cases as affirm would have a high accuracy and would be difficult to beat. In our case, high accuracy doesn’t imply that a model is good one, instead we use f1 score as our metric and use cross validation to verify our scores.

# 3 Featurization

Our goal is to predict whether a district court case will be affirmed or reversed in circuit court by learning on the texts of district court cases. Our input consists of text from district court case and our output is $affirmed$ or $reversed$ in circuit court.

We propose two kinds of models for learning: (1) Models based on n-grams of the text (2) Convolutional Neural Network trained on word embeddings of the text. N-grams are generated using legal context free grammar. Word embeddings help us to represent the text in lower dimensions and to capture multivariate relationship between words and their meanings.

## 3.1 N-Grams

Reversal prediction was mainly based on ngram feature. After filtering out the district cases with information about reversals, we generated 1-4 grams from the text of district cases. We used legal context free grammar for this purpose which is illustrated in Table 1. The idea is to filter the frequencies on the parts of speech sequence to obtain informative key phrases.

Table 1: Context Free Grammar for N-Grams

S ⇒ TWO | THREE | FOUR

TWO ⇒ A N | N N N | . . .

THREE ⇒ N N N | A A N | . . .

FOUR ⇒ N C V N | A N N N | . . .

A ⇒ JJ | JJR | JJS

N ⇒ NN | NNS | NNP | . . .

V ⇒ VB | VBD | VBG | . . .

This ngrams were then filtered based on frequency. We limited our feature set to the top one-eighth most occurring and top one-eight least-occurring ngrams from the total to reduce the dimensionality of the ngram feature, as the middle ngrams are likely to have no effect.**(author?)** [2]. Stopwords and other common court terms were removed from the ngrams. Finally, ngrams were stored into files with their respective frequency to be used later.

## 3.2 Word Embeddings

For convolutional neural networks**(author?)** [8] **(author?)** [15], we generated a word2vec embedding model for court cases using gensim **(author?)** [11] library. Word2Vec model's vocabulary was created on district court training data and was trained on both district court and circuit court texts. Number of dimensions were set to 100 with minimum count of 15 appearances.

## 3.3 Judges Biography

The Circuit Court Judges biography data was the first set of features that were included into the model. We tried to reflect the party affiliations of each judge and also the overall party affinity of the juding panel as a whole. For this we included variables for each judge telling whether they are Republican, Democrat or belong to another party. We also included the ratio of judges in the panel belonging to each party as a normalized score of the representation of each of the parties under consideration.

### 3.4 Case Characteristics

Next, we tried to incorporate important characteristics of the case into the model. We started with adding the variables lower court disposition to indicate whether the case was affirmed or reversed when it was appealed from the district court to the circuit court. Next, we added the feature representing dissent amongst the district court judges while making the decision as we felt that cases with judgment which was not unanimous tend to be better chances of reversal. Finally, we added the issue area under which the case falls under. This was done to reflect the category of cases which might influence reversals. The areas included include 'Criminal Procedure','Civil Rights', 'First Amendment', 'Due Process', 'Privacy', 'Attorneys', 'Unions', 'Economic Activity', 'Judicial Power', 'Federalism', 'Interstate Relations', 'Federal Taxation', 'Private Action' and 'Miscellaneous'. Initially, we added the view of the case as conservative or liberal as a feature but later removed it fearing bias in the original classification.

## 4 Model Building and Training

### 4.1 District to Circuit

Figure 2 provides an overview of the data to analysis pipeline for the district-to-circuit prediction.

After generating the features, we had problem of data imbalance which we solved by applying oversampling to increase weights of reversal cases by duplication and LDA to reduce dimensionality of feature matrix **(author?)** [5] **(author?)** [14], we used ensembles method with gradient boosting and random forest. Finally, we moved on to prediction from convolutional neural networks using word2vec embeddings of district text.

Five convolutional 1D layers representing 1-5 grams with output dimension of 256 were used for classification in case of CNNs. Using 'tanh' as the activation function with l1-l2 regularizer and dropouts to limit overfitting, we trained the CNN on input dimension of word2vec embedding's size for each layer. We finally concatenate and flatten the output followed by application of a dense layer to manipulate its dimensions.

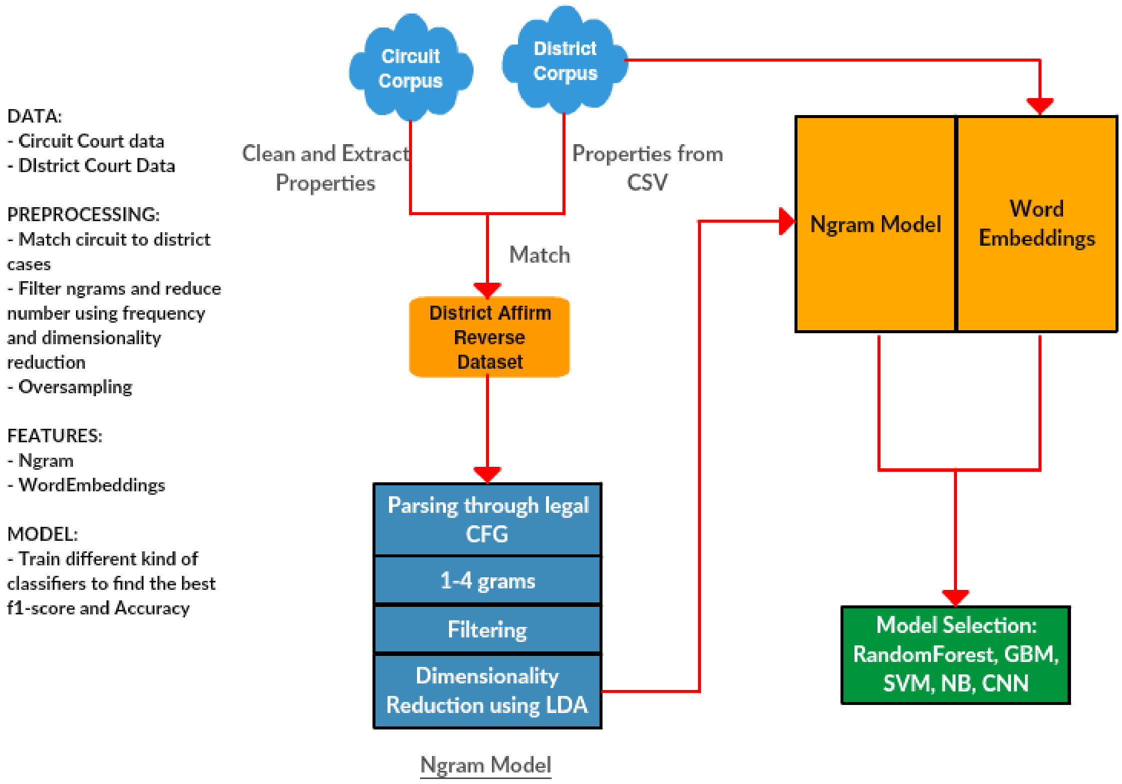

DATA:
- Circuit Court data
- DIstrict Court Data
PREPROCESSING:
- Match circuit to district cases
- Filter ngrams and reduce number using frequency and dimensionality reduction
- Oversampling
FEATURES:
- Ngram
- WordEmbeddings
MODEL:
- Train different kind of classifiers to find the best f1-score and Accuracy
Circuit Corpus
District Corpus
Clean and Extract Properties
Properties from CSV
Match
District Affirm Reverse Dataset
Parsing through legal CFG
1-4 grams
Filtering
Dimensionality Reduction using LDA
Ngram Model
Ngram Model
Word Embeddings
Model Selection: RandomForest, GBM, SVM, NB, CNN

Figure 2: Flowchart shows how the pipeline for predicting reversal in district to circuit goes like. Circuit and district corpus are used to extract ngrams and word embeddings which are then passed into various classifiers for classification

## 4.2 Circuit to Supreme

Figure 4 illustrates the pipeline corresponding to data generation and analysis for the circuit-to-supreme prediction task.

We used the n-gram features to train various classifiers, starting with linear SVM. Looking at results from linear SVM classifiers we concluded that, using n-grams as features its possible to predict the affirms/reversals better than random classifier. Then to improve our predictions we moved to ensemble based classifiers, we trained Random Forest**(author?)** [1] and Gradient boosting**(author?)** [3] with regression trees classifiers. Then instead of using n-grams as features we used Convolutional Neural Networks with Word2Vec embeddings. Using varied metrics to calculate the accuracy we were able to choose the classifier that performed best.

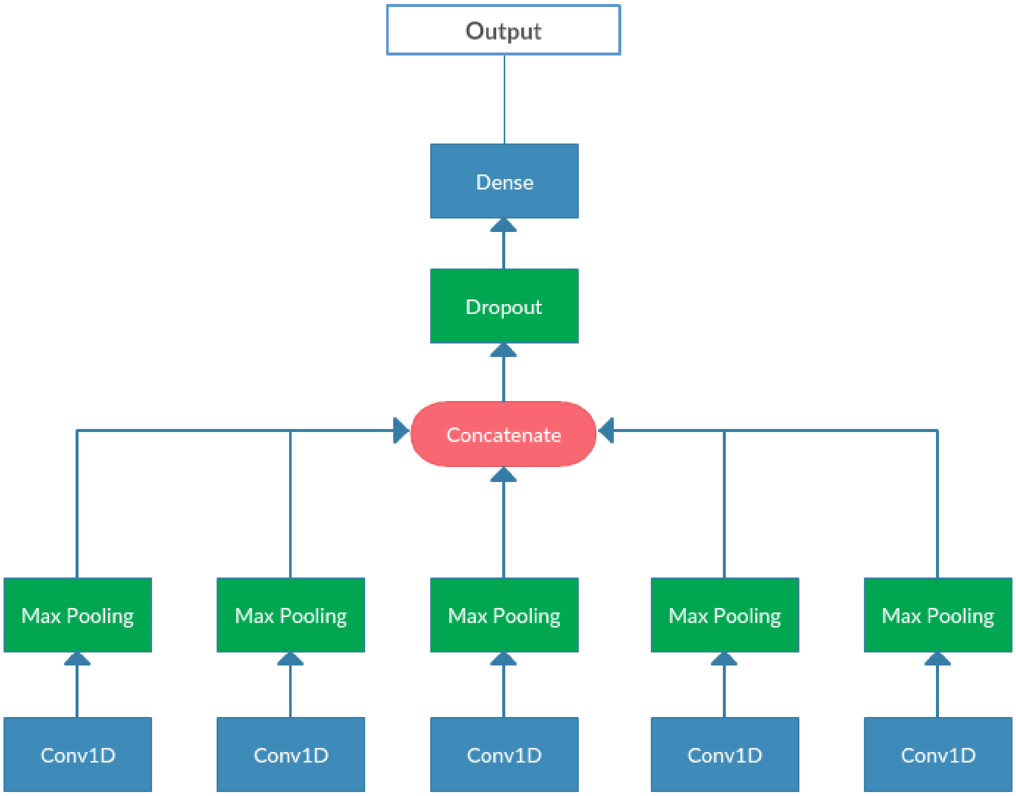


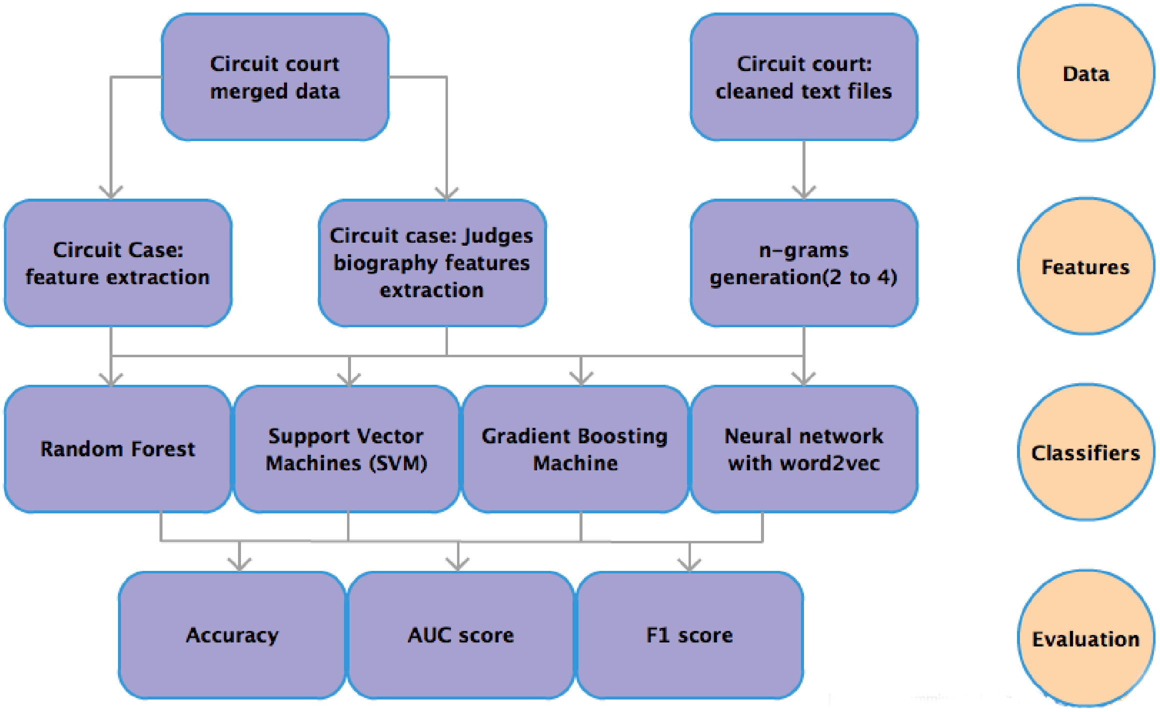
Figure 3: Convolutional Neural Network for case prediction classification. Five 1D convolution layers can be seen for each 1-5 gram which are then max pooled, concatenated and dropped out. CNN is one of the best working classification models we have

We tried two approaches to dataset generation.

1. We built one dataset with the all the cases up to a certain term as the training data and all the cases after the term as the test data.
2. Next, we sampled a proportion of cases from each term (0.75) as training data and rest (0.25) contributed to test data.

We found that using Method 2 for data generation did not yield good results suggesting that future cases might not be a good indicator of reversal of past cases. So we continued with the dividing of data based on particular term for the rest of our analysis, i.e., Method 1.

Figure 4: Pipeline diagram for circuit court to SCOTUS


Circuit court merged data
Circuit court: cleaned text files
Data
Circuit Case: feature extraction
Circuit case: Judges biography features extraction
n-grams generation(2 to 4)
Features
Random Forest
Support Vector Machines (SVM)
Gradient Boosting Machine
Neural network with word2vec
Classifiers
Accuracy
AUC score
F1 score
Evaluation

We had to first vectorize the n-grams before using them as input into sklearn classifiers. For this we used dict vectorizer provided by sklearn. Then these vectorized n-grams and label classes(Affirmed/Reversed) were used as input to various classifiers. While using word2vec model instead of using n-grams, we generated the word embeddings for the text and used this to train our basic convolutional neural network model.

## 5 Results: District to Circuit

The results obtained are tabulated in Table 1. We can see that the best results are obtained using the CNN trained for prediction. CNN produced an accuracy of 0.79 and an F1 score of 0.75, which is much better than the results produced by other classification algorithms. We verified this using cross validation by taking different validation and training sets of 10% and 90% in five folds.

## 5.1 Ngrams

Ngrams are the most important feature used in the methodology. We found that filtering out most of the ngrams based on $\frac{1}{8^{th}}$ strategy doesn't really bring a lot of change in accuracy and f1 score. This proves that only the top most occurring and top least occurring ngrams prove to be valuable features. We also produced a word cloud (Figure 5) using the ngrams which shows that "United States" as an appellee or appellant has a high significance in court decisions. Some words which show emphasis on a statement like "really, control" have greater significance in decision.

| Classifier | F1 Measure | Accuracy |
|---|---|---|
| Gradient Boosting (RT) | 0.70 | 0.76 |
| Multinomial NB | 0.68 | 0.73 |
| Random Forest Classifier | 0.65 | 0.71 |
| LinearSVC Classifier | 0.65 | 0.68 |
| Logistic Regression | 0.67 | 0.70 |
| CNN | 0.75 | 0.79 |

Table 2: Table show F1 scores and Accuracy of all classifiers used. Gradient boosting performs best among the baseline models. CNN perform best overall.

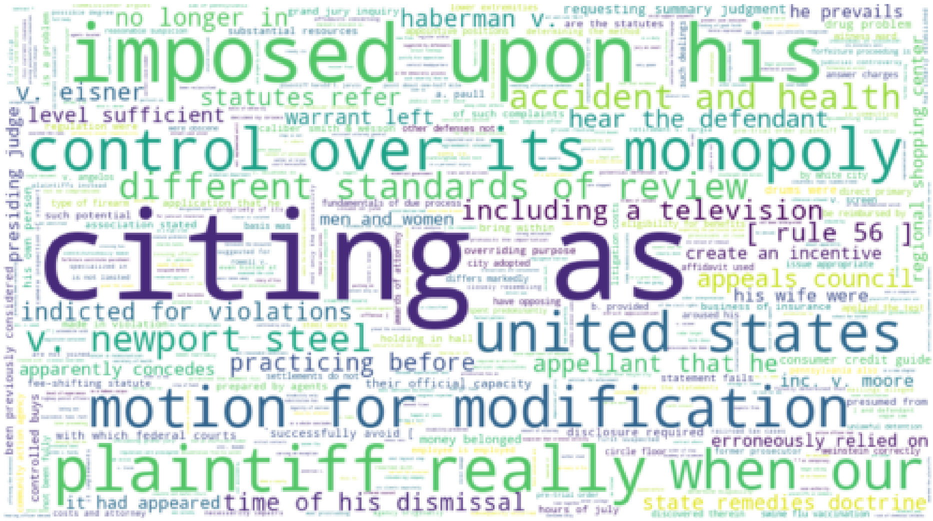


Figure 5: Word cloud showing the most important words affecting reversal of a district case. The bigger the size of the word is in the word cloud, more important the word is in predicting reversals. We can see that words such as "united states" and action verbs which feel intuitive are present in the word cloud

## 5.2 CNN

CNN produced the best results for our study. The graphs below show number of epochs versus various metrics for training and validation data for CNN. With proper regularization, we found that validation loss, accuracy and f1 measure improved as the number of epochs increased. The curve starts to overfit as the number of epochs grow. We prevented overfitting to a limit with dropout and regularization techniques. We also observed that validation loss doesn't always decrease with training loss as there are bumps in between due to different dropout rates.

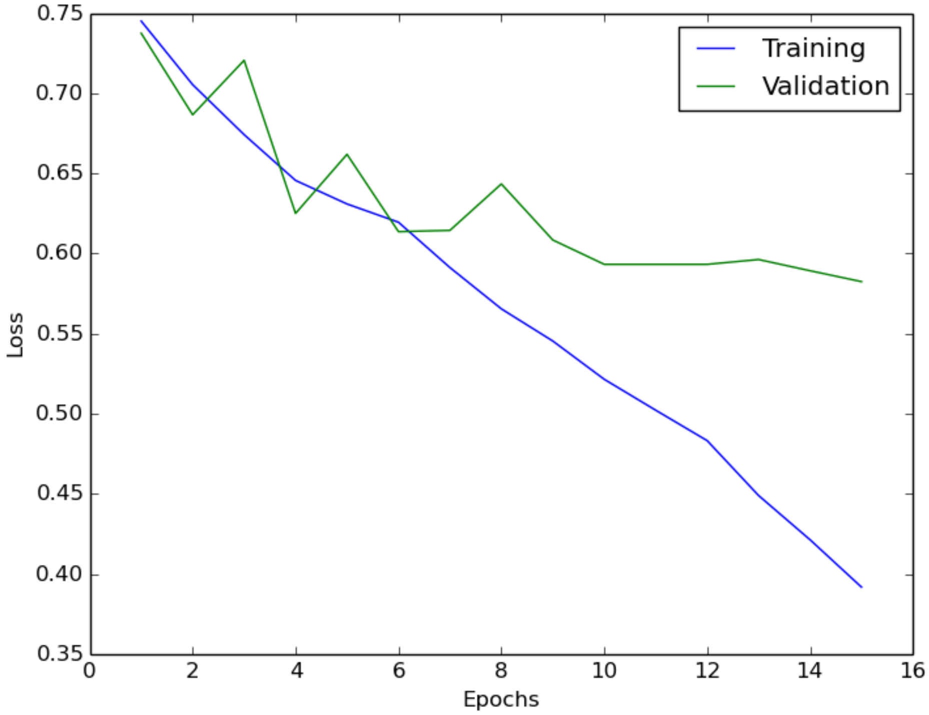


Figure 6: Loss vs Number of epochs in case of CNN for district to circuit reversal prediction. We can see training loss decreases consistently, while validation loss decreases in general with some bumps.

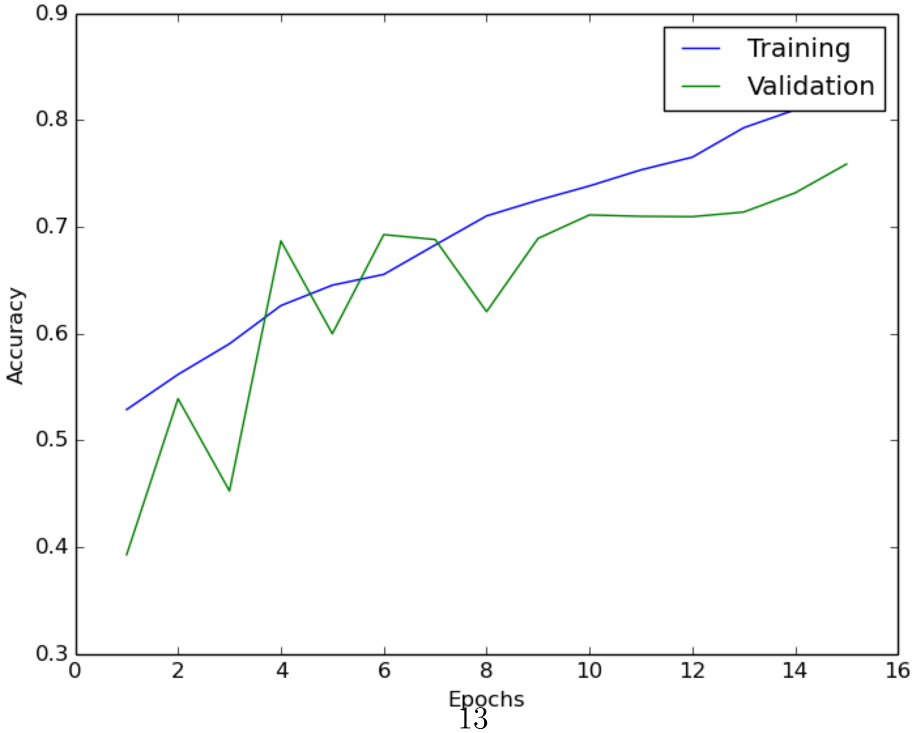


Figure 7: Accuracy vs Number of epochs in case of CNN for district to circuit reversal prediction. Accuracy increases for validation data consistently and starts achieving a constant value as model starts to overfit on training data.

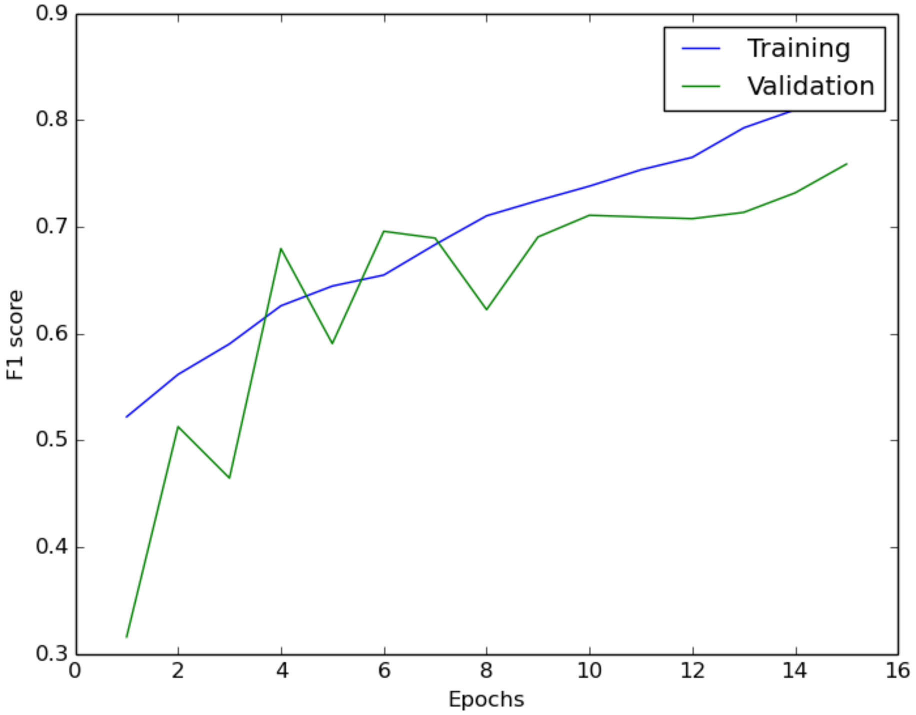


Figure 8: Fmeasure vs Number of epochs in case of CNN for district to circuit reversal prediction. Fmeasure increases for validation data consistently and starts achieving a constant value as model starts to overfit on training data..

# 6 Resuts: Circuit-to-Supreme

| Classifier | Accuracy | AUC | F1 |
|---|---|---|---|
| Random Forest | 63.5 | 0.54 | 0.77 |
| Gradient Boosting | 62.8 | 0.54 | 0.75 |
| SVM | 61.7 | 0.52 | 0.71 |
| Word2Vec | 61.2 | 0.51 | 0.81 |

Table 3: Various Classifier Metrics.

From our experiments we found that Random Forest gives us the best accuracy for the final data set. Though, the base Word2Vec model we implemented gave the best F1 score, the confusion matrix suggested that it was predicting reversed as the outcome for most cases.

| Data Model | Accuracy | AUC | |
|---|---|---|---|
| Case Text | 63.5 | 0.54 | |
| Case Text + Judges Biography | 66.9 | 0.55 | |
| Case Text + Judges Biography + Case Characteristics | 68.1 | 0.56 | |

Table 4: Random Forest Classfier.

We can see an evident improvement in prediction as we add more features into the Random forest Classifier.

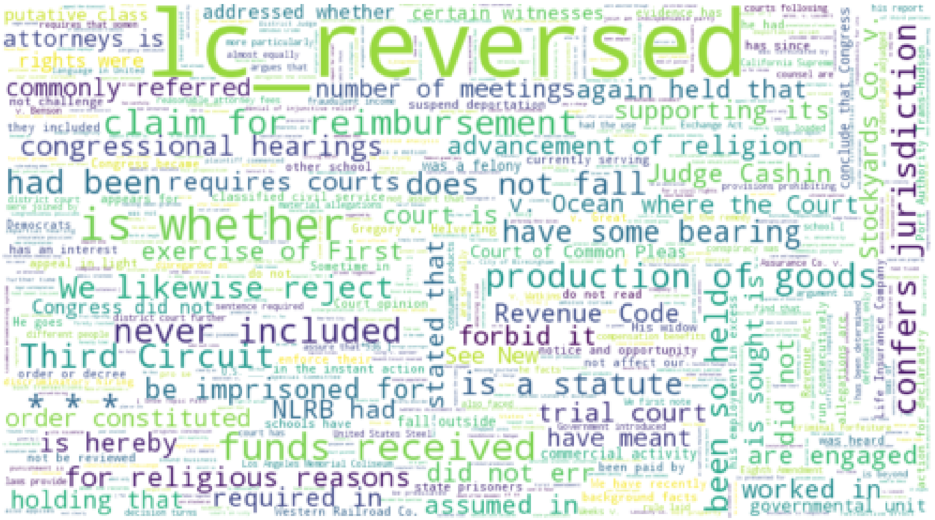


Figure 9: Word Cloud showing Important Features

# 7 Conclusion

## 7.1 Discussion (District to Circuit)

In this project, we tried to predict the reversal of a district court decision by a circuit court using the summarized decision text of the district court. Since, the data was imbalanced, we used oversampling to reduce its effect. We observed that CNN performed the best among all classification techniques used. The results were attained by exhaustive experiments and trials with CNN and many other classification models. From the word cloud obtained, we found that some words in the decision text have a greater effect in the decision such as "United States" as an appellee or an appellant has high significance. We also discovered that some particular action verbs have more impact on the decision of the court.

For improvements, we can train our model with state of the art convolutional neural networks **(author?)** [15] to discover underlying relations which haven't been discovered. The word cloud depicts that citations are an important aspect in decision of a circuit court. Thus, after generating a proper citation graph for district cases, we can merge the citation level affirmed or reversal to our current model. Another important aspect would be to try out similar cases' information. We can create a cluster of similar cases using the text and location. Then, we can add affirmed or reversed information of similar district court cases to our model.

## 7.2 Discussion (Circuit to Supreme)

Examining the results we found that predicting the Supreme court reversals based on the text of the circuit court alone does not give great results. Adding additional features helped to improve the accuracy. From the word cloud we can see that the lower court disposition, i.e, whether the case was affirmed or reversed in the lower court was shown to have a great impact on the prediction scores. The word2vec model without proper regularization and tuning overfits the data as the inherent importance of memes generated through n-grams is not fully captured.

## 7.3 Future Work

Citations is an important aspect which can be incorporated into the model. The citation graph will give the influence of previous cases on current decision and can identify weak precedents. Furthermore ,the Circuit of origin **(author?)** [13] of the case can also be factored into the prediction as historically it has been seen to have an effect. After identifying the important features their weights can be tuned more to give better results. The word2vec model showed much promise in predicting reversals from District to Circuit court cases. Hence, we feel that concentrating on improving its parameters in our basic model should give better results.

Our analysis totally excludes all information that is not available before the case is appealed in the supreme court. Hence the models can make better predictions if features such as the biography of supreme court judges are included. Adding data from the Supreme Court also opens the avenue for other predication such as reversal at the vote level of each of the nine judges. Further, this can be aggregated to predict the dissent between judges in the panel.

# References

[1] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. 4.2

[2] William B Cavnar, John M Trenkle, et al. N-gram-based text categorization. *Ann Arbor MI*, 48113(2):161–175, 1994. 3.1

[3] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001. 4.2

[4] Roger Guimerà and Marta Sales-Pardo. Justice blocks and predictability of us supreme court votes. *PloS one*, 6(11):e27188, 2011. 1

[5] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009. 4.1

[6] Daniel Martin Katz, Michael James Bommarito, and Josh Blackman. Predicting the behavior of the supreme court of the united states: A general approach. 2014. 1

[7] Daniel Martin Katz, Michael J Bommarito II, and Josh Blackman. A general approach for predicting the behavior of the supreme court of the united states. *PloS one*, 12(4):e0174698, 2017. 1

[8] Yoon Kim. Convolutional neural networks for sentence classification. *EMNLP 2014*, 2014. 3.2

[9] Fred Kort. Predicting supreme court decisions mathematically: A quantitative analysis of the âright to counselâ cases. *American Political Science Review*, 51(01):1–12, 1957. 1

[10] Andrew D Martin, Kevin M Quinn, Theodore W Ruger, and Pauline T Kim. Competing approaches to predicting supreme court decision making. *Perspectives on Politics*, 2(04):761–767, 2004. 1

[11] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`. 3.2

[12] Theodore W Ruger, Pauline T Kim, Andrew D Martin, and Kevin M Quinn. The supreme court forecasting project: Legal and political science approaches to predicting supreme court decisionmaking. *Columbia Law Review*, pages 1150–1210, 2004. 1

[13] Stephen Wermiel. Scotus for law students (sponsored by bloomberg law): Scoring the circuits. `http://www.scotusblog.com/2014/06/scotus-for-law-students-sponsored-by-bloomberg-law-scoring-the-circuits/`, Jun. 22, 2014, 10:28 PM. 7.3

[14] Jieping Ye, Ravi Janardan, Qi Li, and Haesun Park. Feature reduction via generalized uncorrelated linear discriminant analysis. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1312–1322, 2006. 4.1

[15] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015. 3.2, 7.1