SMU Data Science Review

Volume 5 | Number 2 Article 2

2021

Qualitative Leveraging Natural Language Processing to Establish Judge Incrimination Statistics to Educate Voters in Re-elections

Aurian Ghaemmaghami
SMU, aghaemmaghami@mail.smu.edu

Paul Huggins
SMU, paulh@mail.smu.edu

Grace Lang
SMU, graciel@mail.smu.edu

Julia Layne
SMU, julia.layne@gmail.com

Robert Slater
SMU, rslater@mail.smu.edu

Qualitative Leveraging Natural Language Processing to Establish Judge Incrimination Statistics to Educate Voters in Re-elections

Aurian Ghaemmaghami; Paul Huggins; Grace Lang;
Julia Layne, Dr. Robert Slater

Masters of Science in Data Science, Southern Methodist University,
Dallas, TX 75275 USA

Abstract. The prevalence of data has given consumers the power to make informed choices based off reviews, ratings, and descriptive statistics. However, when a local judge is coming up for re-election there is not any available data that aids voters in making data-driven decision on their vote. Currently court docket data is stored in text or PDFs with very little uniformity. Scaling the collection of this information could prove to be complicated and tiresome. There is a demand for an automated, intelligent system that can extract and organize useful information from the datasets. This paper covers the process of web scraping and implementing natural language processing (NLP) in order to pull court information and criminal information from public datasets and tie it back to judges. A Condition Random Fields (CRF), Support Vector Machine (SVM), and a Bi-Directional Long Short-term Memory (LSTM) Model were weighed against their predictive accuracy scores to determine the best model in order to tag the dockets for the key entities, or tokens. This paper focuses on the initial keywords that would be beneficial in sentencing trends (ie. name of Judge, defendant lawyer, & state representative). The bi-directional LSTM had the highest accuracy score of 99.4%. This paper will serve as the blueprint for further NLP analysis that will be championed by Code for Tulsa with the possible assistance of other civic groups such as Tulsa Legal Hackers.

1 Introduction

The judicial system is the backbone to the law-and-order system that is set within the United States of America (U.S.). U.S. citizens rely on the democratic election of local and regional judges to uphold the law. Elected judges become the pinnacle of enforcement by rationally issuing sentencing for crimes. However, once a judge is in office, they serve the four-year term with minimal oversight from the public on judgments during their term. As judges come up for re-election or face a new challenger for their seat, there is no data available on how they treated and sentenced crimes within their term.

Currently, judge conviction rates, sentencing lengths, and releases in the State of Oklahoma are not easily accessible to the public. While the data is public, if a voter

wants to understand if a judge up for re-election is particularly relaxed on drug related crime, they would need to visit multiple state-run sites and specifically know the criminal defendant's name in order to view the judgment description. Even knowing this, there could be variation on where that information would appear in the court web pages because each transcriber of those court dockets writes down the sentencing information differently. Building out a full view of a judge's moral stances on subjects would be tedious and subjective to uncover. Uncovering sentencing trends by conviction type would be even more difficult to assess quantitatively.

According to a personal conversation with a retired Texas judge, Judge Ron Champion, it is very difficult for the general public to get a full picture of a judge-elect due to lack of any political campaigning (2021). The foundation of a judge's role is to approach each individual case unbiased, weigh the evidence, and fairly assign a punishment. Campaigning with a distinct platform of pro-gun ownership or legalization of marijuana may place the judge in a biased light with the public, so campaigning is typically frowned upon and not very prevalent. Yet on the contrary, the judges end up selecting a political party affiliation before entering their name on an election ballot.

This contradictory situation ends up leaving the general population with little knowledge about the judge's character. Many times, during an election period a brief biography may be posted online for a judge-elect to establish reputation. However, once elected, these campaigns or positions are removed from websites so that the judge may be impartial to the public. There are some sites that have local officials listed year-round, but they only include contact information at most (League of Women Voters, 2021).

Public judgment data at the judge level is not currently available. Ballotpedia.org has election results for judges: who has been elected and by how many votes. However, this does not tell a common voter why a particular judge may be the best person for the job.

Voter turnout for Oklahoma is 55%, which is vastly under the national average of 66.4% (Blatt, 2020). It is difficult to conclude why the voter turnout is so much lower than the rest of the U.S.; however, one sentiment that could add to this non-action is: why vote if I do not know who I am voting for?

The aim of this study is to develop an interface to properly inform regional voters in Oklahoma of judges' judgment towards crime and provide insight into sentencing trends over time by conviction type. The extracted data would grant voters the bi-partisan control to make an educated decision on their local leadership, as well as support local legislature in possible sentencing reform. Code for Tulsa would upkeep the data by the automation of the data scraping and extracting process.

2 Background

Natural Language Processing (NLP) is a subset of machine learning and programmatically helps computers manipulate and process human language. Named Entity Recognition (NER) is a subset of NLP and will be used in this study for evaluating a corpus of open court data. Other machine learning techniques that will be explored are support vector machines (SVMs) and neural networks.

This paper will determine the highest accuracy of the various Natural Language Processing (NLP) models using the OSCN and ODOC public datasets.

2.1 Named Entity Recognition Approach

Reviewing the Oklahoma State Court Network (OSCN) and Oklahoma Department of Corrections (ODOC) datasets can be cumbersome and difficult to extract information as each docket is transcribed differently and not every court docket is complete. A similar NLP work effort was conducted in Pakistan on court judgments leveraging Named Entity Recognition (NER) (Iftikhar, 2019). Iftikhar, Ul Qounain Jaffry, & Malik described how they tapped into the potential of text data from legal proceedings, and how they utilized machine learning models in order to extract the relevant data for various trend analyses. Named entity recognition is the process to locate and classify Named Entities (NEs) from text into pre-defined categories such as names, case numbers, locations, quantities, etc. NER is used in the practice of this analysis to answer questions, such as:

1. Which judges were involved in the hearing/sentencing events?
2. What crime was the criminal charged for?
3. How long was the sentence?
4. Is there an average sentence by conviction that could be used for legislature guidelines? (Currently there is not legislature in place for these types of guidelines with Oklahoma judges)
5. Was there any chance of probation or appeal?
6. Was a plea deal offered? How common are plea deals with certain convictions?

An example of a court docket is as follows (Figure 1).

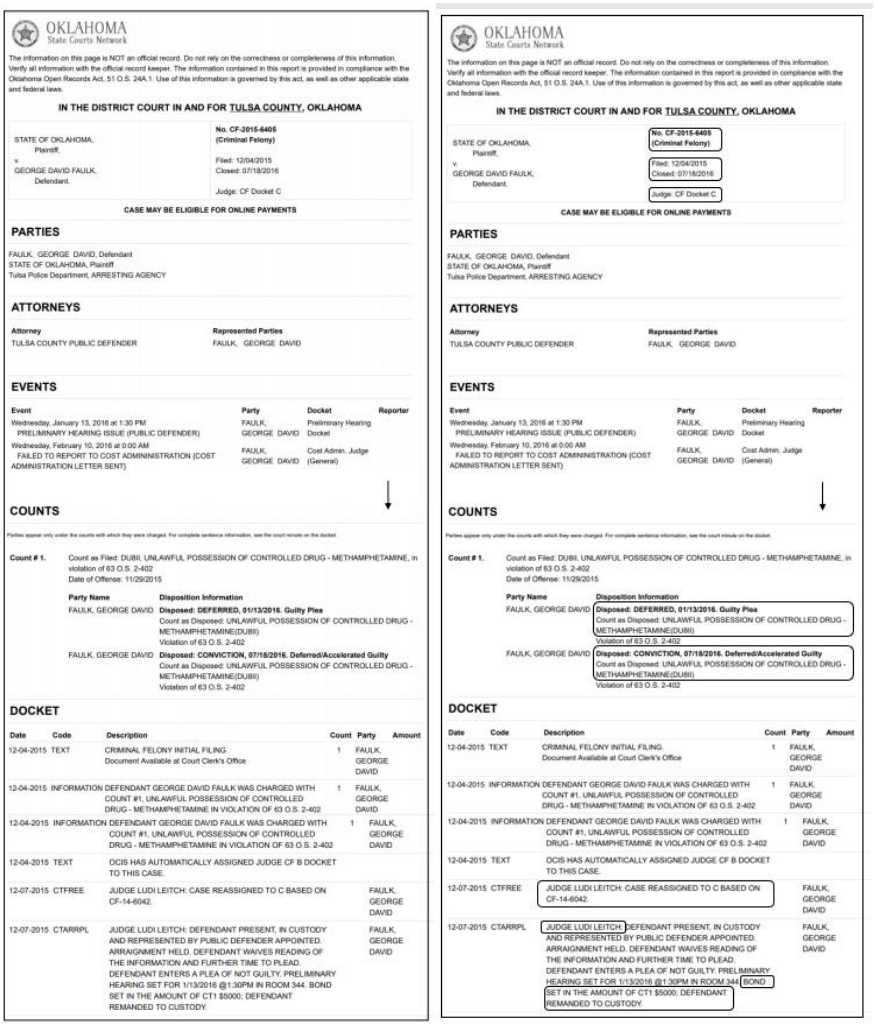


Fig. 1. The version on the right side has the areas within the docket that would serve to be beneficial for the analysis. Some of these circled areas would be considered named entities in which the model could train and classify based on the identities given.

In order to train the model to recognize and predict judge names that are written within the docket, a sample set will be fed into the model that shows examples of how a judge's name can be written and how to classify it. For example, a judge making a sentencing for a conviction might be written as "JUDGE MOODY" or "JUDGE DAWN MOODY". Both versions are used within the NER model for the model to train and look within the sequence of the tagging. NER models use past data, such as the sample set fed into the training model, in order to anticipate how future data will be classified.

NER will also be used in order to test a training sample set utilizing HTML code. A JSON version of the OSCN data scraped from the website could potentially be used to identify conviction sentencing and judges when it is marked in a royal blue color. The HTML code can be tokenized and classified just like any other named entities in order to train the model to best find the fields needed for this analysis.

2.2 Neural Networks

When it comes to Natural Language Processing, Recurrent Neural Networks (RNN) is a popular type of network used in textual data processing. The RNN structure contains an internal loop that processes sequential data one step at a time instead of trying to process the entire data at once. This is achieved by an internal "memory" network within the RNN structure where it takes into consideration the previous textual data word by word. In a more condensed version, RNN's will read one word at a time and slowly start building an understanding of what the entire textual data means. This type of structure is crucial in this research paper due to the flexibility of the recurrent "memory" layer which will allow the model to learn patterns efficiently (Graves, et al., 2013). The "memory" portion can be explained by Figure 2 below (Torti, et al., 2019).

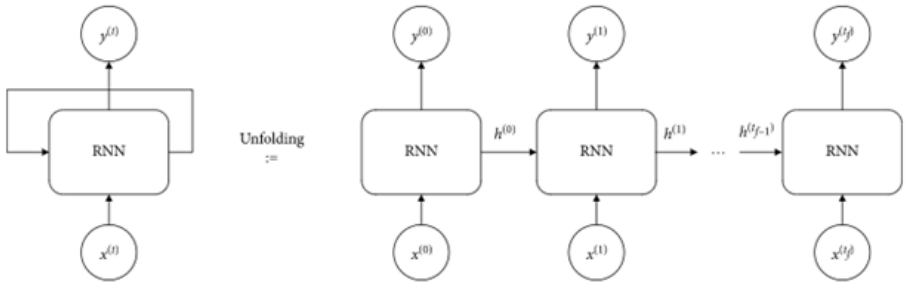


Figure 2 (Torti, et al., 2019): RNN network structure.

The first word the recurrent layer sees is x^0 and will translate the output to be y^0 at that specific point in time. Then x^1 utilizes both the second word and the output of y^0 to formulate the y^1 output. This is how the RNN internal memory structure operates to slowly develop an understanding of the textual data. This research utilizes sequences of words closer to 1000 characters. The RNN starts to see diminishing returns once the text reaches a certain threshold of characters. The reason for this is because it gets difficult for the model to build the text in general once the sequence gets long. The model then has trouble trying to remember what it has seen at the beginning because it is now insignificant due to all the outputs it has seen since then (Graves, et al., 2013).

Long short-term memory (LSTM) is an extension of the recurrent layer that solves the memory pitfalls described in the RNN structure prior. In the RNN layer, the internal state memory was only tracking the previous output. By the time the model reaches x^2 it loses the output of x^0 because the output of x^0 is stored in the memory and combined

with the output of x^1. LSTM models can access the output from any previous state at any point in the future. This adds more complexity to the model for sequences that get very long because it is easy to forget things in the beginning (Chiu, et al., 2016). LSTM addresses that memory issue by keeping track of text seen at the beginning or in-between sequences. What this means for this research is that the LSTM can look anywhere at a point in time which is crucial for identifying key entities within the OSCN dockets.

Prior studies highlight the application of document classification systems using Recurring Neural Networks (RNN) and Bidirectional Long Short-Term Memory networks (Bi-LSTM). Given court dockets are in the form of sequential text format, the utilization of Bi-LSTM's is most favored amongst the variety of RNN architectures (Lukasiewicz, Petrova, and Armour, 2020). Bidirectional LSTM's read in the given inputs twice in both forward and reverse order and store those in memory via an additional embedding layer. It is essentially combining two independent RNN's together which creates a final concatenated output allowing the network to understand words in a sentence more efficiently for prediction purposes. Figure 3 illustrates the Bi-LSTM network in detail (Cornegruta, et al., 2016).

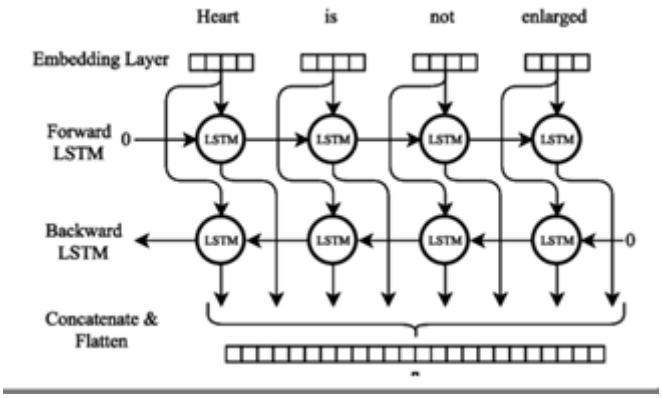


Fig. 3. Bi-LSTM network design (Cornegruta, et al., 2016)

The literature from Lukasiewicz, Petrova, and Armour (2020) is helpful to the success of this research since these models will provide an optimal recognition pattern that will aid in locating accurate judge nomenclature. Similar research has been done in other countries to improve their own court information retrieval system. Braz et al. (2018) utilized Bi-LSTM models to handle the unstructured irregularities in their scanned court documentation system (Braz, et al., 2018). The final model allowed their research team to effectively identify over 84% of new supreme court dockets to the correct associated party (Braz, et al., 2018). Currently, inmate information in Oklahoma is not tied directly to a sentencing judge. These findings provide a further level of confidence in the applicability and diversity of Bi-LSTM model in named entity recognition-based settings of unstructured data. One aspect of this research is to tie in both findings of these studies to provide an un-bias model capable of accurate judge and inmate classification.

Given that this is the space of deep learning, one method that is used for enhancing the computational capabilities of text classification is conditional random fields (CRF) (Jasmir, et al., 2021). CRF is a probabilistic model that aims to predict a set of labels corresponding to a given sequence of text inputs (Z. Wan, et al., 2019). One milestone of this research is to measure the precision and recall scores of the named entities we are trying to tag (defendant lawyer, judge, and state rep). Below is a representation of how the CRF model works behind the scenes (Jasmir, et al., 2021).

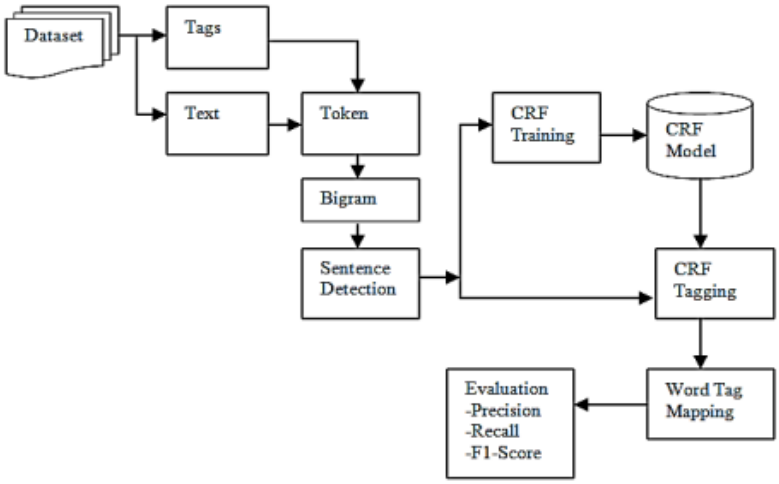


Figure 4: CRF base architecture (Jasmir, et al., 2021).

The CRF process is essentially taking the given training data and parsing them into tags/text which are then transformed into "tokens" for text pre-processing. The output is then the trained model with associating labels of the given features (Jasmir, et al., 2021). These final outputs are combined features that will be utilized in assessing the level of importance in a text sequence (Jasmir, et al., 2021). This is a crucial method within the research that will help significantly increase the accuracy of our text classification techniques.

As stated in the application of these models in a similar judicial setting, the objective of this research is to leverage a Bi-LSTM algorithm to effectively capture accurate sentencing judge information from the court docket inputs fed into it. As highlighted in the previous section detailing named entity text extraction, naming judge entity inputs prior to curating an RNN, Bi-LSTM model is crucial to maintaining the integrity of the data nomenclature. Curating standards and rules for entity-based nomenclature is key when trying to create a high performing neural network. Given the unstructured nature of the OSCN data, a Bi-LSTM model's feature of reading inputs backwards and forwards is the level of robustness that is needed when trying to identify sentencing judges within the court dockets.

2.3 Support Vector Machines

Support Vector Machine (SVM) methods have been used to distinguish patterns within textual documents in a legal context in prior studies. A study done by Medvedeva, et. Al (2020) focused on predicting verdicts from relatively unstructured European court case documents. The model digested a portion of the court documents as training data and then proceeded to break down the text within the documents using NLP. Once the data was trained, it was given a set of testing data to apply the algorithms on. The testing data was of similar textual layout to the training data except it had the final verdict removed from the document. The study utilized a custom computer program that was designed to analyze ECtHR (European Court of Human Rights) case documents and predict a verdict (Medvedeva, 2020). The program split the case data into three groups: facts, arguments and decisions. In the end, the facts were used to predict the ultimate decisions (Medvedeva, 2020). SVM's aim to create a hyperplane that separates data points into identifiable groups based on features within the data. The simplest algorithm resulting in the lowest amount of error is defined as the final SVM model.

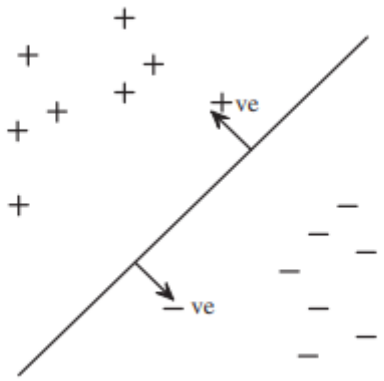


Fig 5. Visual representation of the plane created in SVM modeling. (Campbell, 2011)

The SVM model was trained using the training data and then given the testing dataset to measure the performance and achieve an accuracy metrics based on the percentage of correctly identified verdicts. K-fold cross validation was used to determine optimal parameters for the final model (Medvedeva, 2020). The final model had an overall accuracy of 75% and similar average precision and recall scores. The study noted that higher overall metrics could be achieved given more training data and a balanced training set. The distribution of verdicts was not equal, and the performance metrics were dampened by the uncommon verdicts being falsely predicted. Utilizing the SVM approach in the context of document text classification and sentencing patterns could be beneficial in defining labels and features that hold a higher importance when it comes to judge sentencing pattern association.

SVM's were used in a similar study done on Chinese legal documents to determine verdict patterns in divorce cases (Li, 2019). This study is unique in its approach as it compared the effects of support vector machines alongside of neural networks to assess the viability of each option in accurately predicting outcomes. Bi-LSTM and CRF were chosen as the neural network algorithms to compare against. The results concluded that the neural network was able to outperform the SVM in this study by roughly 6% in an F1 score (Li, 2019). It was determined that the semantics of the legal documents were better suited for neural networks given the complex nature of capturing and recognizing underlying context in the dockets. Another drawback of using SVM's in NLP problems is scalability. The dataset that this study utilizes contains thousands of documents with each document having a potential length of over 1,000 words that could potentially pose computational problems for the model. The sheer amount of training data required to fully train an acceptable model is a time-consuming task that might not scale well for this type of project. SVM's compute a dimensional plane between datasets, wherein multiple planes would need to be calculated for each document. This necessity for multiple dimensional planes is not only computationally challenging, but also time consuming.

These studies highlight how support vector machines have been used in court document specific studies focused on both verdict prediction and pattern recognition. While the neural network garnered slightly better metrics when compared head-to-head with the SVM, it demonstrates the flexibility of both methods and opens the door for testing both methods in future work. The aim of this analytical study is to compare bi-directional LSTM, SVM and RNN to help best extract entities from court dockets for further trend analysis.

3 Methods

The collection of data through computers enables more digital data to become public knowledge within the government sector. Every court documented event, fee, and charge of an individual is now captured and recorded on the Oklahoma State Court Network (OSCN) site. OSCN data has insight into defendant name, felony charges, bond amounts, judgments and sentencing to name a few. This data lists any document or process filed in court as a text string. Likewise, the Oklahoma Department of Corrections (ODOC) captures data about the criminal defendant. The ODOC data includes text data on criminal's name, age, sex, race, court of conviction, sentencing length, current jailing status, and offense. Merging the two datasets is critical to understanding judge sentencing trends.

The court dockets that are used in this research are available on OSCN's website. A python scraper developed by Code for Tulsa was used as the first step of this analysis in order to extract the text into a format easier to mine (Dungan, 2020). However, the scraper pulls the text into run-on fields without designating any categories or identifiers to the data. The scope of this analysis was limited to criminal felony cases in Tulsa County during the years 2012 to 2020. The Tulsa court had around 46,000 criminal felony cases during that time. These dockets have on average around 1,000 tokens each. A token is defined as an individual word, character (such as punctuation), or a sequence of characters that are "grouped together as a useful semantic unit for processing" (Manning, Raghavan, & Schütze, 2008).

This study will use Named Entity Recognition (NER) to identify a set of tokens as the sentencing judge on a document. A named entity is a set of tokens that represent a real-world object, place, person and anything else that might be considered a proper noun in grammar. The goal of NER is to extract out all entities from a document or section of text.

NER takes in a tokenized list of words and marks each word as the beginning of an entity (B), continuation of an entity (I), or not related to an entity (O). The only entities to be tagged in the training set of documents will be the name of the Judge associated with the sentencing. All other tokens will be marked as 'O'.

The first task in creating this tagged data set is to identify 200 example court cases within OSCN and to manually copy out the sentencing judge's name into a CSV file with the associated case number. This data set was manually created by the authors by referencing court case numbers from the ODOC data set and by using the web interface for OSCN. The small training set limits the amount of too much variation for the initial study of identifying sentencing judges. Even within this subset, there was still variation by year and court docket.

From this tagged training set of identified sentencing judges and court case numbers, the OSCN Python package created by Code for Tulsa was used to scrape the web page

of the associated court case dockets and retrieve all HTML of the page (Dungan, 2020). The case docket objects were broken into individual sections. The text from each section was then combined into the full text of the court documents. The full court document text was tokenized by spaces and punctuation before developing a word map of unique words, or tokens, that are used throughout the training dataset.

To account for judges with initials in names and the potential importance of the `:` colon identified in manual tagging, special characters were not removed after tokenization of the text.

The final tokenized version of the document was reviewed to find tokens identified as the sentencing judge entity via regular expressions. Every instance of the judge's name that matches the pattern identified near sentencing was marked as an entity in the document and all other tokens are marked as other, `O`.

Padding is the process of making each sentence, or in this case docket, the same length. Neural networks require to have all sentences inputted into the model to be uniform in length (Shrestha, 2020). Padding was used at a document level to ensure that each docket is of the same length, considering some case dockets are written with more text and court filings than others. Each document was scanned for length and fed into a histogram to be evaluated. The histogram was analyzed to determine optimal document length to ensure that roughly 95% of the documents were full length. This process removes outlier documents that may be in the 5th percentile for longest text. The documents that are shorter than the cutoff value was padded by adding '0' tokens to fill the space at the beginning of the document.

With tokenization, padding, and the initial tagging of the dataset complete, the tagged documents were fed into a Recurrent Neural Network (RNN). This RNN will receive the text two ways: first with left to right reading and second in reverse order, right to left reading. This is to allow the RNN to learn potential importance of the tokens leading up to an entity and those tokens that come after.

3.0 Exploratory Analysis

3.1 Data Sources

As described in the methodology, the two main datasets this analysis leverages are the OSCN and the ODOC public datasets for this research. The OSCN dataset includes sentencing judge information, type of charges against the defendant, sentencing length and associated court fees. The data comes split into three different tables outlining unique information about the defendant regarding the defendant's personal profile, sentencing/probation lengths, and offense committed. It is important to first merge these three tables together within ODOC to get the full defendant profile prior to joining it into the OSCN dataset. To do this, there are unique identifiers for each inmate denoted as a document number identifier. Once complete, the two independent ODOC and OSCN data sources are ready to be joined together so the analysis can accurately tie

back the sentencing judge with the associated inmate. However, it is important to note the hurdles and inaccuracies when trying to join these two datasets together.

Unfortunately, there is no primary key identifier in the OSCN dataset that would intuitively tie back to the ODOC inmate data. Instead, this research utilized another unique column within the ODOC dataset that shares the same structural properties as the court docket number associated with a certain case. However, the textual elements within this column needed to be cleaned further to join back accurately to the OSCN dataset. String manipulation techniques were leveraged to create a uniform structure amongst the textual elements that helped tie the information back seamlessly. Following the data merge, there were several egregious text-based nuances for several of the older cases within the ODOC dataset dating back prior to 2006. These cases did not match back up correctly with the OSCN information. Due to the nature of this research, as noted in the background, the scope was limited to only include cases from 2012-2020 onward to adjust for this noise. There are a couple of key reasons that led to this decision, but the main one is the inconsistencies within older court docket cases. The OSCN database is sensitive to inmates who have had cases prior to 2012. This assumption is that this may be an encoding issue from Oklahoma's Corrections Network since the irregularities of the data fields primarily happen on dates greater than ten years. Restricting the range alleviates those issues and curates a workable dataset ready for exploration.

Court cases present in both the OSCN and ODOC databases in the year 2021 pose a unique challenge of their own. Critical docket information that can tie the two datasets back together is missing from recent entries because these cases are still being processed in their early stages. Lastly, judge information is not present for some of the more recent cases because the docket system has not randomly assigned a judge to the case. Cases that fall under this category will be dropped from the model until a judge is assigned, and the RNN can extract this judge label for analysis. As the case information is updated, the cases will be added back into the model and newer cases without this information will be dropped.

3.2 Data Cleaning

A critical part of this research's data cleaning efforts is understanding that much of this text-based data from OSCN and ODOC is human entered, which means there can be many inconsistencies within several of the key attributes disrupting the integrity of the RNN model. Due to this, it is very important to maintain the basis of each feature to the best ability without much manipulation since most of this is personal data. One large hurdle needed to overcome was managing the date fields within the ODOC datasets post-merge. There were columns that denoted the day an inmate was sentenced or charged with his/her crime and the last time an inmate was moved within or between other jail facilities. There were various strings with random inputs containing field values such as "00:00:00" or "JR". It was crucial to create a set of helper functions to

go through the entire ODOC dataset and reformat the dates to the proper "YYYY-MM-DD" schema. Out of 1,069,028 records, about 250 records fell victim to this date entry error. Because of this, the analysis will move forward without those records as it can cause problems when trying to cross-reference it back to a sentencing judge.

Further investigation within the ODOC date fields as described above led to another discovery. The analysis uncovered multiple birth date fields that were more recent than the field when an inmate was last moved between facilities. So, why is that important for this research? It means there are errors within these fields because it is impossible for someone to be born after their move date. Luckily, only 6 out of the 1,068,778 records were guilty of this data entry error. Those records were filtered out of the dataset moving forward to ensure a heightened level of robustness for the modeling efforts.

3.3 Handling Missing Values

When it comes to handling personal data, it is very important to understand the ethics around imputing and dealing with missing data entries. The researchers in this analysis needed to approach the problem objectively before making hasty decisions on dropping data columns or removing null fields. For example, there are height and race ODOC fields with null values. Imputing the null values of height with the mean value might cause a dilemma where there is too much of an assumption of someone's height based off other humans. Each human is their own individual and treating them as such is an integral part of the research design to maintain data consistency and integrity. If height is imputed with a general mean, then the analysis would be generalizing groups of people from different populations as one, and that is too egregious of an assumption to run a model on. Same logic applies when dealing with how to impute missing values for the race attribute. A researcher cannot simply impute someone's race based off the frequency of other races in a dataset. Because of these reasons, this analysis has refrained from dropping missing data for any personal attributes of a given convict within the ODOC dataset.

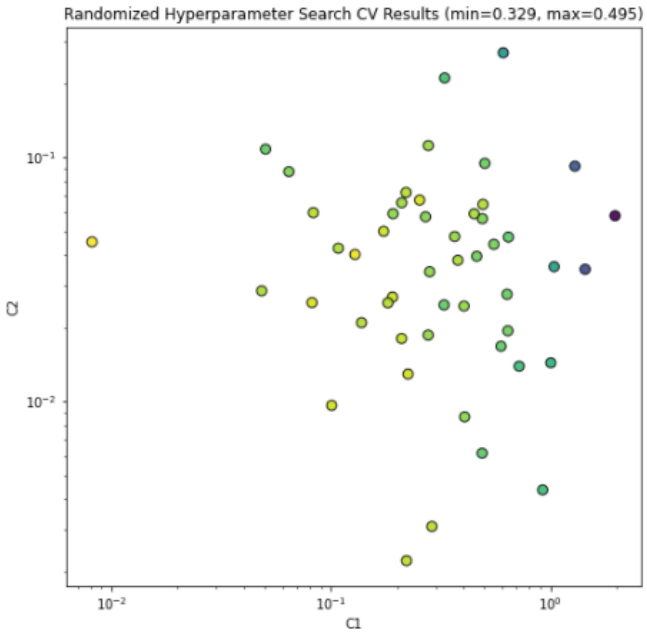
However, lengthy investigations of the ODOC dataset have led to an interesting correlation between the court docket number, statute code, sentencing county and sentencing date fields. Whenever a missing value is found in one of these fields, the other corresponding three fields are also missing. After investigating several records, there is strong evidence to suggest these inmates don't carry any criminal felonies or drug-related crimes. Since the scope of the data is focused on those types of crimes, removing the 121,920 entries that fall victim to the predicament explained above is crucial for this research design.

Lastly, there were some features dropped due to the impracticality of the attribute. Features like "suffix" and "Unnamed: 3" were fields that either weren't telling a story important enough to include in the model features or the feature was inexplainable. "Unnamed: 3" for example had only 47 records that were non-null with the input of

"Y". The assumption was that these records had incomplete first and last names; however, the 47 records all had applicable names and there wasn't much information on the ODOC website to decipher the use case. Due to these inconsistencies, the column was dropped from consideration.

4.0 Results

A baseline Conditional Random Fields (CRF) model was run to allow for the comparison between multiple models. The CRF model achieved a precision score of 0.800 for the B-Judge token and 0.739 precision for the I-Judge token. The recall scores were 0.421 and 0.395 respectively. The weighted average F1-score was 0.566. This model serves solely as a baseline to improve upon.



	precision	recall	f1-score	support
B-Defendant lawyer	0.818	0.439	0.571	41
I-Defendant lawyer	0.793	0.523	0.630	44
B-State Rep	0.783	0.439	0.562	38
I-State Rep	0.783	0.439	0.562	41
B-Judge	0.800	0.421	0.552	41
I-Judge	0.739	0.395	0.515	43
Avg	0.786	0.443	0.565	248

Fig. 6. Randomized Grid Search with Cross Validation results showing optimal values for C1 and C2 to be used in the CRF model. Confusion matrix for the final CRF model.

The padding methodology revealed that 75th percentile of document length was 736 words, the 95th percentile was 965 words and the 99th percentile was 1,120 words. A value of 1,000 was used as the max length and all documents under this threshold were padded to achieve this length.

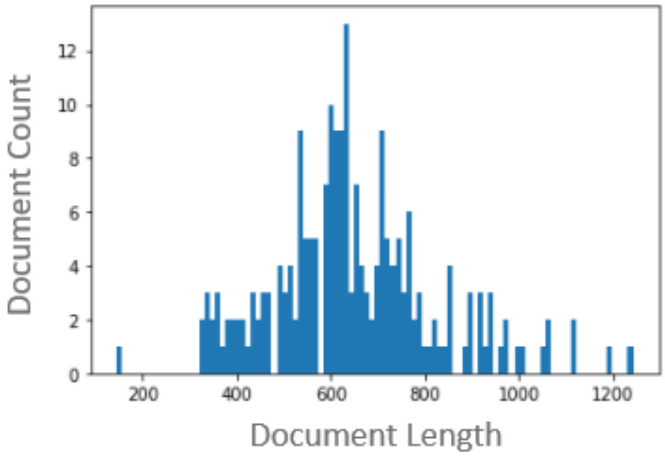


Fig. 7. Padding is a process to create homogeneous text strings for the model to process simultaneously. The histogram above highlights the varying document lengths, for the researcher to choose the proper cutoff to pad the documents while still maintaining performance.

The dataset consisted primarily of the 'O' class and needed to be weighted to properly score the model and provide reliable output. Weighting was used to force the 'I' class into weighing more heavily when accuracy metrics were calculated. As an example, if a sample of the data contained 99 instances of the 'O' class and only 1 instance of the 'I' class, the model needs to weight the 1 'I' class more heavily than the other 99 'O' class. In this vein, the weighting process was used to provide more reliable accuracy metrics by weighting the tags 2000 to 1 for all non NER tags.

Tuning of the LSTM model included a word vector length for word embeddings of 300 which was manageable to run, but still gave reasonable results. The learning rate and drop rates were tuned for 0.25, 0.35, 0.05 and 0.01, 0.05, 0.005 respectively. The overall accuracy was evaluated from a five-fold cross validation to determine the most optimal options for the final model. Based on the results the higher dropout rates and lower learning rates had the best performance. From these results, a dropout rate of 0.35 and a learning rate of 0.01 were chosen for the final model.

Drop Out Rate			Learning Rate			
0.25	0.35	0.5	0.005	0.01	0.05	Accuracy
		x			x	42.491
	x				x	43.987
x					x	86.180
x				x		99.980
		x		x		99.988
x			x			99.989
	x		x			99.990
		x	x			99.992
	x			x		99.995

Fig 8. Ablation Study of Drop Out Rate and Learning Rate. Accuracy from running a 5-fold cross validation over an LSTM Neural Network with the given parameters.

From these settings, the final model was run against the full 200 tagged dataset using a 5-fold cross validation to get the accuracy, precision and recall.

	precision	recall	f1-score	support
B-Defendant lawyer	0.500	0.025	0.048	40
I-Defendant lawyer	0.300	0.120	0.171	50
B-State Rep	0.500	0.100	0.167	40
I-State Rep	0.438	0.175	0.250	40
B-Judge	0.500	0.200	0.286	40
I-Judge	0.692	0.205	0.316	44
Avg	0.488	0.138	0.206	254

Fig 9. Classification Report of the Final LSTM Model.
Tags 0 and 6 are padding and non-NER fields respectively.

This final LSTM model had an overall accuracy of 0.994. The average precision and recall for NER tags were 0.488 and 0.1375 respectively. Whereas the non-NER tags had an average precision and recall of 0.9955 and 0.9995 respectively. Though low, the precision is higher than recall for the NER tags. In this case when trading off between precision and recall, precision is the preferred metric. It is better to be sure a tag is correct than mis-tagging someone as the Judge in the case. Incorrectly associating someone with a case is a higher cost to that person than not having that entity tagged.

Based on these results, machine learning is a viable option for identifying judges and other named entities in court documents. The CRF is the better model at identifying named entities in the scraped web text from OSCN, despite the length of the text

processed. It was also faster to train and could make for cheaper processing time in production usage.

5.0 Discussion

With this 99.4% accurate bi-directional LSTM model Code for Tulsa can continue the work effort needed to pull and extract information for OSCN data with assistance from groups such as Tulsa Legal Hackers. This process will be able to provide a dataset that a front-end developer can leverage in creating a UI reference for judge sentencing in Tulsa once it is connected to the ODOC dataset.

When taking the final LSTM model and running it against a small set of case numbers in the ODOC dataset from 2019, trends in incorrect tagging drifted toward overly tagging the defendant as one of the selected NER tags. When building onto the test dataset, the defendant and any other frequently mentioned names should be considered for tagging so that the model can learn to differentiate between this highly frequent token and other named entities.

The dataset will need to be expanded to improve upon the accuracy, precision, and recall of any model this gets tagged against. With thousands of cases a year in Tulsa County alone, this small set of 200 only makes up a small fraction of the total cases on the dockets every year.

The pipeline of labeling named entities in a spreadsheet and string matching added many tags. Using a tagging tool could be a more user-friendly way to build a larger dataset and limit the instances of names we care about. This would require further increase in the weighting of classes, as the ratio would be even more disproportionate. However, it would focus the model in proximity to the sentencing verbiage where Code for Tulsa will look to pull out further information.

5.1 Challenges

Some of the obstacles faced during this analysis were limiting scope, training the data using different NLP techniques (bidirectional LSTM, RNN, etc.) and cross validation techniques. Initially the analysis was to join one unstructured data set (OSCN) to a structured dataset (ODOC) by developing a key in the NLP process in order to extract information for analysis. During the exploratory data analysis phase, it became clear that the effort needed to train and extract the key entities from OSCN would be very time intensive. The analysis scope zeroed in on just the OSCN dataset and extracting key entities from the docket, such as Judge, Defendant's lawyer, and the State’s representative lawyer. The exploratory data analysis that was executed on the ODOC dataset still cleaned and prepared for any additional future work to join the two tables together.

Model bias can occur when training with text that shows up more frequently than others. In this analysis identifying a Judge name was the ideal outcome of the model, but there are only around 10 judges that served on the criminal courts during 2012 – 2019, which the model might pick up on if it was only getting trained on judge name. Including other key names, such as state representative and the defendant's lawyer, helped mitigate any model bias. Cross-validation techniques were also used in order to act as a secondary wave of eliminating model bias. A k-folds technique was used in model training and testing.

5.2 Ethics

The goal of the use of this data is to be able to extract and represent macro trends within the Oklahoma court network. Some of the members who work as attorneys for Oklahoma state expressed the desire to have public reports of these trends like the Annual Statistical Report the state of Texas publishes each year (Texas Judicial Branch, 2020).

The purpose of the data use is not to single out that Judge A is lighter on sentencing times for marijuana cases than Judge B or to draw conclusions that Judge A is a better judge because of this data statistic. Even though this data is public record, it does contain personally identifiable information. The results still must be treated ethically with the understanding that drawing any conclusions on a judge could affect their livelihood. What the data will not depict is the unique circumstances and qualitative details that a judge will hear from the defendant in court in-person. These qualitative aspects of case will never truly be represented within the quantitative forms of the data that is pulled from this analysis.

Along with not showing any judge bias, it is key to accurately depict if there are any racial trends by sentence within the data. A study completed in 2004 in state felony courts demonstrated that "black and Hispanic defendants tend to receive harsher sentences than white defendants" (Demuth). On the contrary, in 2019 Indiana University investigated racial bias in the context of judicial decision-making and found that "results showed White probationers at low-risk levels received longer sentences relative to Black probationers classified at the same risk levels" (NewsRX LLC). Representing this data in the most accurate and unbiased nature would be key to the model ethics.

In the next phase of this analysis, it will be critical to ensure that the proper ethical steps are taken in order to not bias the data in its presentation to the public. For example, when the front-end user interface is developed by the Code for Tulsa, they will need to develop visualizations and results that keep the data as statistically based as possible without using language or designs that would draw conclusions on how a judge will sentence based on a particular conviction. It would not be fair to generalize how

sentencing will unfold for a defendant when there are always extenuating circumstances that make each individual case unique in the judgement process.

6.0 Deployment

The next stage of this analysis the model will be scaling out to all available case dockets on the OSCN network. In 2018, there were around 5,700 criminal felonies in Tulsa County, 6,100 cases in 2019, and about 5,800 in 2020. There are 77 counties in the state of Oklahoma. While the other counties may not be the same scale of criminal felony cases as Tulsa County, ensuring that the OSCN scraper and NLP model is scalable will be critical to the success of the database creation. Building the initial historical database back to 2012 will take some processing time and capacity from the Code for Tulsa team; however, once the initial database is built with prior years, the scraper and model will only need to run on a monthly basis in order keep the database updates to a minimum. The database and scraper would not need to run more often because it takes some time for criminal cases to make their way through the court system. The database updates could also benefit to be aligned with the ODOC database, so that once a connection is created between the two datasets, there is less data gaps to explain in future analysis. As described previously, ensuring a judge has been assigned to a specific case will also be what the model will look for in future implementation.

Tulsa Legal Hackers communicated that they had strong interest in being able to identify if there were any trends with sentencing guidelines by conviction. A member of the group recently published an article on the need for sentencing reform in the state of Oklahoma (McCarty, 2021). McCarty describes how "Oklahoma currently has some of the longest sentences in the world...[which] is due to an outdated criminal code" that provides extremely long ranges of punishment. Oklahoma also enforces an 85% rule on all incarcerations, which requires all convicts to serve at least eighty-five percent of their prison sentence before becoming eligible for parole. The long sentences and the 85% rule can be costly to taxpayers without directly correlating to a reduction in crime. With better descriptive statistics of sentencing within Oklahoma, civic groups could develop a case to begin the initial steps of reforming sentencing practices that saves taxpayers dollars without having an increase in imprisonment rates.

Code for Tulsa has expressed interest in building out more of the NLP model in the next phase to answer more questions, such as:

1. The District attorneys' policies on who they give plea deals to are not very transparent. Are there any trends in those plea cases that could answer how plea deal discretion is used?
2. How do sentencing lengths differ across Oklahoma counties?
3. Are there any racial trends in sentencing length by conviction type?

7.0 Directions for Future Research

There are many other systemic issues that will not be directly answered with the scope of this analysis, but could be considered in the future use of this work:

- How is a new voter supposed to understand who may be the best candidate?
- Are the judges currently in office fairly distributing sentencing based on judgments given?
- How does a voter educate themselves on a challenging judge's principles versus the incumbent judge?
- Are there any sentencing trends by conviction type that could be used to understand discretion guidelines of judges?
- Trends by political affiliation. A study conducted in 2019 found that "Republican-appointed judges sentence black defendants to 3.0 more months than similar nonblacks and female defendants...compared to Democratic-appointed judges" (Cohen & Yang).

8.0 Conclusion

The prevalence of unstructured data outweighs the amount of structured data, especially in the judicial system. This analysis serves as the first step towards the state of Oklahoma being able to utilize and analyze macro sentencing trends while being able to drill down to the county, judge and potentially conviction type. Utilizing a bi-directional LSTM served to be the best fit model in extracting entities out of the unstructured docket data. This research could serve as model for other states to implement something similar for states wanting to be able to build upon any descriptive statistics from their court documents.

References

1. Blatt, D. (2020, November 15). Oklahoma is dead last in voting participation, and that's not good enough. Retrieved March 7, 2021, from https://tulsaworld.com/opinion/columnists/david-blatt-oklahoma-is-dead-last-in-voting-participation-and-thats-not-good-enough/article_20dfcdd4-243d-11eb-a0dc-9f42c945a9b8.html
2. Braz, F. A., Correia da Silva, N., Borges, F., & Inazawa, P. H. (2018). Document classification using a Bi-LSTM to unclog Brazil's supreme court. CoRR - Information Retrieval (Cs.IR), abs/1811.11569, 1–5. https://dblp.org/rec/journals/corr/abs-1811-11569.bib
3. Campbell, C., & Ying, Y. (2011). Learning with support vector machines. Morgan & Claypool. https://doi.org/10.2200/S00324ED1V01Y201102AIM010
4. Champion, Ron. Personal communications – telephone. April 6, 2021.
5. Chiu, J. P., & Nichols, E. (2016). Named Entity Recognition with Bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, *4*, 357–370. https://doi.org/10.1162/tacl_a_00104

6. Cohen, A., & Yang, C. S. (2019). Judicial Politics and Sentencing Decisions. American Economic Journal: Economic Policy, 11(1), 160–191. https://doi.org/10.1257/pol.20170329
7. Cornegruta, S., Bakewell, R., Withey, S., & Montana, G. (2016). Modelling Radiological Language with Bidirectional Long Short-Term Memory Networks. *Proceedings of the Seventh International Workshop on Health Text Mining and Information Analysis*. Published. https://doi.org/10.18653/v1/w16-6103
8. Demuth, S. (2004). Ethnicity Effects on Sentence Outcomes in Large Urban Courts: Comparisons Among White, Black, and Hispanic Defendants. Social Science Quarterly, 85(4), 994–1011. https://doi.org/10.1111/j.0038-4941.2004.00255.x
9. Dungan, J. (2020, September 5). OSCN utilities. Retrieved March 7, 2021, from https://github.com/codefortulsa/oscn
10. Findings from Indiana University Update Understanding of Criminal Behavior (Racial Bias and LSI-R Assessments in Probation Sentencing and Outcomes). (2019). In Politics & Government Business (p. 60–). NewsRX LLC. https://link.gale.com/apps/doc/A574465103/ITOF?u=txshracd2548&sid=bookmark-ITOF&xid=9c10b5d1
11. Graves, Alan; Mohamed, Abdel-rahman; & Hinton, Geoffrey. (2013). Speech Recognition with Deep Recurrent Neural Networks. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, (pages 6645–6649)
12. Iftikhar, U. (2019). Information Mining From Criminal Judgments of Lahore High Court. IEEE Access, 7, 59539–59547. https://doi.org/10.1109/ACCESS.2019.2915352
13. Jasmir, J., Nurmaini, S., Malik, R. F., & Tutuko, B. (2021). Bigram feature extraction and conditional random fields model to improve text classification clinical trial document. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, *19*(3), 886. https://doi.org/10.12928/telkomnika.v19i3.18357 .
14. League of Women Voters. (n.d.). *Educating Voters*. Educating Voters | League of Women Voters. https://www.lwv.org/elections/educating-voters.
15. Li, L. (2019). Research and Design on Cognitive Computing Framework for Predicting Judicial Decisions. Journal of Signal Processing Systems, 91(10), 1159–1167. https://doi.org/10.1007/s11265-018-1429-9
16. Lukasiewicz, T., Petrova, A., & Armour, J. (2020). Extracting outcomes from appellate decisions in US State Courts. 133–142.
17. Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Tokenization. https://nlp.stanford.edu/IR-book/html/htmledition/tokenization-1.html.
18. McCarty, C. (2021, January 12). *Guiding Principles of Sentencing Reform*. Oklahoma Justice Reform. https://okjusticereform.org/guiding-principles-of-sentencing-reform/.
19. Medvedeva, V. (2020). Using machine learning to predict decisions of the European Court of Human Rights. Artificial Intelligence and Law, 28(2), 237–266. https://doi.org/10.1007/s10506-019-09255-y
20. Shrestha, S. (2020, July 7). *NLP: Preparing text for deep learning model using TensorFlow2*. Medium. https://towardsdatascience.com/nlp-preparing-text-for-deep-learning-model-using-tensorflow2-461428138657.
21. *Statistics & Other Data*. Texas Judicial Branch Seal. (2020). http://www.txcourts.gov/statistics/annual-statistical-reports/.
22. Torti, E., Musci, M., Guareschi, F., Leporati, F., & Piastra, M. (2019). Deep Recurrent Neural Networks for Edge Monitoring of Personal Risk and Warning Situations. *Scientific Programming*, *2019*, 1–10. https://doi.org/10.1155/2019/9135196

23. *The Texas Politics Project*. Texas Politics - Crime and Punishment in Texas: Statutory Wrongdoing and Its Consequences. (n.d.). https://texaspolitics.utexas.edu/archive/html/just/features/0201_01/crimeandp.html.
24. Wan, Z.; Xie, J.; Zhang, W.; & Huang, Z. (2019). "BiLSTM-CRF Chinese Named Entity Recognition Model with Attention Mechanism," J. Phys. Conf. Ser., vol. 1302, no. 3