The Supreme Court: Issue Area and Disposition Classification through Fine-tuned Language Models

Haikal Rozaidi, Maitreya Dixit , Abhinay Putta and Dylan Yang
rozaidi.1@osu.edu
dixit.80@osu.edu
putta.12@osu.edu
yang.6294@osu.edu

Abstract

Classifying legal documents accurately is crucial for efficient case management and legal research. Traditional manual classification is time-consuming, error-prone, and expensive. This project proposes fine-tuning existing NLP models to better understand the legal language in Supreme Court petitions to allow automated categorization. We focus on categorizing two aspects of a petition: its issue area, and its final disposition. We leverage data sets like SPAETH and Comparative Agendas, for ground truth labels, supplemented with full-text petitions from the Supreme Court website and the ProQuest database, for textual analysis. Our ultimate goal is to generate a model that is able to truly understand the legal language contained within Supreme Court petitions, as general models are unable to do so. We aim to contribute to NLP applications in the legal domain, improving efficiency in legal text classification.

1 Introduction

The United States Supreme Court receives thousands of petitions each year, all of which differ in their issue areas. Some petitions concern momentous events, like corporate malfeasance, while others focus on less important events, like divorce. Currently, the Supreme Court does not have an in-house department to categorize these petitions by issue area. Instead, this task is usually taken up by other researchers.

Groups like SPAETH and the Comparative Agendas Project have attempted to categorize a subset of Supreme Court petitions based on a topic list that they themselves created (Spaeth, 2024) (Jones et al., 2023). To accomplish the task, researchers in political science and law would manually read through each of the petitions, assigning them a topic based on what they interpreted. Not only is this process extremely time-consuming and expensive, it is also heavily prone to human error.

The Supreme Court petitions are also categorized by their disposition - the Justices final ruling on the case (affirmed, reversed, dismissed, etc.). The dispositions for each petition are instantly made available following a ruling and are organized in publicly accessible datasets, like SPAETH. As such, categorizing the dispositions take significantly less manpower than categorizing the issue areas. However, the court usually takes about 3 to 9 months to grant a ruling after a petition is heard. Finding methods in predicting the ruling of a case solely by the petition filed would be useful in helping petitioners decide if they should continue fighting for their case or not. Analyzing the previous dispositions could also reveal patterns in petitions that favor a certain ruling.

A novel solution to both these problems would be to employ the use of language models to do the categorizations for us. However, general language models today have a difficult time understanding the legal language behind Supreme Court documents. This is because these models are often trained on corpus that focus on regular speak. As such, we are fine-tune existing models to better understand the language contained within Supreme Court petitions to better perform classification tasks especially in issue area and disposition.

2 Datasets

SPAETH: The SPAETH dataset categorizes Supreme Court petitions to their respective issue area and disposition. The issue area list is composed entirely by the SPAETH research group. This data set will act as our ground truth when training the model (Spaeth, 2024).

Comparative Agendas: Similarly, this data set categorizes the petitions according to their respective topic. The topic list is entirely composed by the Comparative Agendas group. While we will be focusing analysis on the SPAETH dataset, we will

potentially use this data set as our ground truth for a future run of the model (Jones et al., 2023).

Supreme Court Website: This site contains the full-text pdfs of Supreme Court petitions from years 2017 to 2024 (Supreme Court, 2024). We will be scraping the pdfs from this website so that we can run the language models on the full-text data.

ProQuest Insight: ProQuest provides access to full-text petitions from years 1946 to 2024 for only paid petitions (ProQuest, 2024). This dataset provides us with a wider set of years. However, paid petitions only make up around 30% of petitions. We will use this dataset in conjunction with the supreme court website dataset, seeing as ProQuest covers the years missing: 1946 to 2016.

3 Prior Work

A core feature of LLM's is analyzing and sorting through large amounts of text, so it is reasonable to assume that other researchers are actively applying LLM's to improve documents for Supreme Court and other legal review. Jonathan H. Choi writes a paper on how state-of-the-art LLM's like GPT-4 can help analyze legal documents while involving Supreme Court opinions in context (Choi, 2023).

However, one main problem arises: specific research on Supreme Court filings is limited, and quality datasets are not publicly maintained to the full extent. Thus, it becomes difficult to fine-tune an LLM for the specific Supreme Court review use case. A recent study by Deroy et al. (2024) explores the use of large language models for summarizing legal case judgments, highlighting both their potential and current limitations.

But then again, we see the authors mentioning the presence of inconsistencies and hallucinations in the outputs of the generative models, which in part comes from lack of rich data for fine-tuning purposes. As a result, our project aims to resolve this issue by first collecting quality data and then fine-tuning newer LLMs with such data.

A relevant study exploring fine-tuning within the legal domain is LEGAL-BERT: The Muppets Straight Out of Law School (Chalkidis et al., 2020). This study explored using BERT as it is, adapting BERT with domain-specific corpora, and pre-training BERT from scratch with public legislation, contracts, and then fine-tuning them. The pre-trained and adapted BERT outperformed the generic model, highlighting the importance of fine-tuning and specialized training to achieve better performance in specific legal applications.

NeoBERT advertises itself as a next-generation encoder that incorporates state-of-the-art advancements in NLP research and outperforms regular BERT models (Le Breton, 2025). Since legalBERT is trained on the regular BERT model, we also aim to fine-tune both legalBERT and NeoBERT on our set of supreme court petitions and compare the results obtained from both models. We also fine-tuned the regular BERT model on the set of petitions as a baseline comparison to both NeoBERT and legalBERT

4 Methodology

We were able to separately fine-tune a NeoBERT, a legalBERT, and a BERT model to the petitions scraped from the full-text datasets (ProQuest and supremecourt.gov). The models were trained on the SPAETH issue areas and the dispositions found in the SPAETH dataset. Our methods can be seen below:

4.1 Data Scraping

To get our exhaustive list of supreme court petitions, we referred to the "Journal" section under the supreme court website (The Supreme Court of the United States, 2024). Each petition is identified by a unique identifier: the docket number. We then filtered the list of docket numbers so that we could obtain a list containing solely petitions that had SPAETH issue areas. This step required cross-referencing both the Journal and the SPAETH datasets.

Our main sources of the full-text petitions are the supreme court website and the ProQuest database. We implemented a python script that would search up, download, organize, and convert to text each petition obtained. Crawling through the web pages and downloading the pdfs was done using selenium. Converting the pdfs to machine-readable text was done using PyPDF2.

After the scraper was done running, our data was stored in a parquet file, with each row representing one petition linked to its SPAETH issue area and disposition.

4.2 Data Preprocessing

The raw data obtained from web scraping was loaded into a csv file containing the case number, disposition, issue area, and full text of the

pdf. We used scipy's train_test_split function to split the full-text dataset into 60-20-20 train-dev-test splits. The training, development, and testing datasets were then saved into parquet files to be used in the model.

The full-text is then preprocessed by removing stopwords and special characters using the nltk library.

Before using the data for finetuning, each split is loaded into pytorch DataLoaders. To do so, the full texts are tokenized using the respective tokenizers of BERT, Legal-BERT, and NeoBERT. Once tokenized and loaded into DataLoaders, the data can be batched and used for training.

4.3 Handling Token Limits

Our dataset contains 3 main features for each data sample. The issue area and case disposition are just integers, so handling these features is not a problem. However, the full text, indicating the entire petition itself, is very large, often ranging from 60,000 to 75,000 tokens of English text. Thus, we came up with two strategies to feed this large amount of text to our finetuning process.

The first strategy was simply taking the first 512 tokens of the petition text for BERT and Legal-BERT, and the first 4096 tokens for NeoBERT. These numbers signify the respective context window limits for each model. This strategy was chosen because the first bit of text contained important header information for each petition. This seemed like an appropriate set of text that could help the model learn classification between issue areas and case dispositions.

The second strategy was first chunking the entire petition text into chunks of 512 tokens for BERT and LegalBERT, and chunks of 4096 tokens for NeoBERT. Then, using the corresponding model's tokenizer, we generate embeddings for each of these chunks. Finally, we average the embeddings of all chunks. Thus, we end up with an averaged embedding representation that fits within the model's context window limit. The averaging was done using PyTorch's inbuilt mean function.

4.4 The Model

We use the pretrained base models 'bert-base-uncased', 'nlpaueb/legal-bert-base-uncased', and 'chandar-lab/NeoBERT' respectively for BERT, Legal-BERT, and NeoBERT respectively. On top of the base architecture, we added a linear layer to the end of the model to make disposition or issue area predictions from the output of the language model. Additionally, a dropout layer with dropout rate of 0.1 was added after the base model's output.

For both issue area and case disposition prediction, Cross Entropy Loss is used to find loss for multi class classification. Total loss is used for weight updates and calculated as the sum of disposition loss and issue areas loss.

For Legal-BERT and BERT, batch sizes of 16 were used and for NeoBERT a batch size of x was used. For all models a learning rate of 1e-3 was used for finetuning. Training was done in 100 epochs.

5 Results

The training, development, and test set accuracies for all three models in predicting issue area and case disposition can be seen below.

5.1 Chunking Tokens Approach

	Training Accuracy (%)	Development Accuracy(%)	Test Accuracy (%)
Issue Area	52.38	55.84	54.40
Case Disposition	35.58	32.48	33.83

Table 1: BERT Accuracy Results with Chunking

	Training Accuracy (%)	Development Accuracy(%)	Test Accuracy (%)
Issue Area	62.34	65.84	63.01
Case Disposition	36.63	36.93	35.41

Table 2: legalBERT Accuracy Results with Chunking

	Training Accuracy (%)	Development Accuracy(%)	Test Accuracy (%)
Issue Area	52.34	55.45	53.71
Case Disposition	34.06	29.90	31.75

Table 3: NeoBERT Accuracy Results with Chunking

Test Set	BERT Accuracy (%)	legalBERT Accuracy(%)	NeoBERT Accuracy (%)
Issue Area	54.40	**63.01**	53.71
Case Disposition	33.81	**35.41**	31.75

Table 4: Model Accuracy Comparisons with Chunking

5.2 First N Tokens Approach

	Training Accuracy (%)	Development Accuracy(%)	Test Accuracy (%)
Issue Area	37.95	42.08	38.18
Case Disposition	34.49	30.10	32.54

Table 5: BERT Accuracy Results for first 512 Tokens

	Training Accuracy (%)	Development Accuracy(%)	Test Accuracy (%)
Issue Area	62.38	66.63	64.59
Case Disposition	36.14	35.94	35.01

Table 6: legalBERT Accuracy Results for first 512 Tokens

	Training Accuracy (%)	Development Accuracy(%)	Test Accuracy (%)
Issue Area	49.80	51.09	51.83
Case Disposition	36.07	31.98	33.93

Table 7: NeoBERT Accuracy Results for first 4096 Tokens

Test Set	BERT Accuracy (%)	legalBERT Accuracy(%)	NeoBERT Accuracy (%)
Issue Area	38.18	**64.59**	51.83
Case Disposition	32.54	**35.01**	33.93

Table 8: Model Accuracy Comparisons with First N Tokens

5.3 Comparing Per-Class Accuracies

	LegalBERT	NeoBERT	BERT
Criminal Procedure	85.00	76.92	96.54
Civil Rights	79.43	60.00	50.29
First Amendment	59.21	60.53	26.32
Due Process	21.95	9.76	0
Privacy	65.22	34.78	13.04
Attorneys	22.73	9.09	0
Unions	70.37	66.67	62.96
Economic Activity	68.85	46.45	69.4
Judicial Power	19.20	60.80	16.8
Federalism	26.09	0.00	0
Interstate Relations	NA	NA	NA
Federal Taxation	80.77	88.46	53.85
Misc.	14.29	57.14	0
Private Action	NA	NA	NA

Table 9: Per Class Issue Area Accuracies when Averaging Embeddings Across Models

	LegalBERT	NeoBERT	BERT
Criminal Procedure	91.54	81.54	86.15
Civil Rights	61.14	58.86	5.14
First Amendment	71.05	44.74	1.32
Due Process	19.51	2.44	0.00
Privacy	78.26	13.04	0.00
Attorneys	31.82	9.09	0.00
Unions	74.07	66.67	3.70
Economic Activity	53.01	28.96	57.92
Judicial Power	35.20	63.20	34.40
Federalism	50.00	0.00	0.00
Interstate Relations	NA	NA	NA
Federal Taxation	69.23	69.23	3.85
Misc.	28.57	14.29	14.29
Private Action	NA	NA	NA

Table 10: Per Class Issue Area Accuracies with the First N Tokens

5.4 Final Approaches Comparison

The tables below show comparisons between the chunking approach and the first N tokens approach for each model, highlighting which approach achieved better test accuracies.

Disp Code	LegalBERT	NeoBERT	BERT
1	NA	NA	NA
2	29.90	81.99	13.83
3	0.52	9.95	3.66
4	81.99	9.32	90.99
5	0.00	21.71	0.00
6	0.00	0.00	0.00
7	0.00	12.90	0.00
8	0.00	0.00	0.00
9	0.00	9.09	0.00
10	NA	NA	NA

Table 11: Per Class Disposition Accuracies with Average Embeddings. *Note: Refer to Appendix 10.1 for Disposition Key*

Disp Code	LegalBERT	NeoBERT	BERT
1	NA	NA	NA
2	26.69	59.81	0.64
3	37.17	38.74	9.42
4	54.97	18.01	95.96
5	3.10	7.75	0.00
6	0.00	50.00	0.00
7	0.00	0.00	0.00
8	0.00	0.00	0.00
9	0.00	0.00	0.00
10	NA	NA	0.00

Table 12: Per Class Disposition Accuracies with First N Tokens. *Note: Refer to Appendix 10.1 for Disposition Key*

	BERT model with first 512 Tokens (%)	BERT model with chunking approach(%)
Issue Area	38.18	**54.40**
Case Disposition	32.54	**33.83**

Table 13: Approach Comparison for BERT

	legalBERT model with first 512 Tokens (%)	legalBERT model with chunking approach(%)
Issue Area	**64.59**	63.01
Case Disposition	35.01	**35.41**

Table 14: Approach Comparison for legalBERT

	neoBERT model with first 4096 Tokens (%)	neoBERT model with chunking approach(%)
Issue Area	51.83	**53.71**
Case Disposition	**33.93**	31.75

Table 15: Approach Comparison for neoBERT

6 Analysis and Discussion

The main points of comparison are in the form of comparing the three models (BERT, legalBERT, and neoBERT) as well as comparing the two approaches (First N Tokens and Chunking).

6.1 Model Comparison

In terms of issue area prediction, legalBERT yielded the best accuracy results when compared to neoBERT and BERT (see Table 4 and 8). This is due to the fact that legalBERT is already pretrained on legal language contained within legal documents in the US as well as in Europe. The results suggest that legalBERT is able to better understand the legal language within the Supreme Court petitions and thus is able to generate issue area predictions that best describe the petition.

Despite neoBERT being advertised as a better version of BERT, the issue area accuracies for both these models are relatively the same. This is due to both models only being trained on regular English. These models are less able to understand the legal language in the Supreme Court petitions.

Case disposition prediction accuracies remained low and consistent among all three models. These models only try to understand the semantic meaning of a document and the relatively low accuracies suggest that the semantic meaning of a document alone is not enough to predict the dispositions. This

suggests that there are other external factors that have a say in a judge's ruling of a case.

6.2 Approach Comparison

In general, it seems that chunking and first N tokens approach yield interchangeable accuracy results with one notable exception (see Tables 13, 14, 15). Chunking the inputs for the BERT model increased it's accuracy by around 16 percent from 38.18% to 54.40% (see Table 13).

This could indicate that the first N tokens of a petition contain similar enough semantic information to the chunked embeddings.

6.3 Issue Area Accuracy Comparison

The issue area of "Criminal Procedure" achieved the highest accuracies for all models. This could indicate that the models are trained on mostly criminal procedure petitions which skew the training data quite a bit (See Table 9 and 10). The model does accurate work in predicting criminal procedure because it has a greater understanding of these petitions from a larger dataset.

"Due Process" achieves consistently low accuracies across both approaches of handling token limits, due to the petition set low distributions of due process documents. Similarly, "Federalism" also scores low in accuracies except for legalBERT (see Table 10).

There were also no examples of Interstate Relations and Private Actions after multiple random samples of test data so the performance of the models could not be tested on these issue areas. More examples of these areas would be needed to make accurate predictions in the future.

6.4 Disposition Accuracy Comparison

The models performed better in predicting disposition codes 2 (Affirmed) and 4 (Reversed and Remanded), followed by 3 (Reversed) than the other codes. This is likely due to their being more instances of these codes in the data for the model to learn from. The accuracies are still generally lower and there is noticeable variance between the different models and the same models using different handling of long text. More training data is likely needed to make improvements in this area.

Multiple case disposition codes also had accuracies of 0% between all models. This is likely because the few documents that existed for these codes might have had a language similar to the case dispositions that were more frequent in the data and got misclassified.

There were also no examples of stay, petition, or motion granted (disposition code 1) and certification to or from a lower court (disposition code 2) after multiple random samples of test data. The performance of our models could not be tested on these disposition codes, and more examples of these areas would be needed for more accurate predictions.

7 Conclusion

LegalBERT proved to be better at predicting issue areas in Supreme Court petitions when compared to neoBERT and BERT. However, predicting case dispositions seems to be a task that legalBERT, neoBERT, and BERT struggle at indicating more external factors are at play than just the meaning of a petition when it comes to a case's final ruling.

The chunking approach when compared to first N tokens approach yields similar results indicating that the first N tokens contain similar semantic meaning to chunked embeddings.

BERT, legalBERT, and neoBERT collectively had widely varying per class accuracies for issue area and case disposition. This is mainly attributed to the skew of the dataset, where some categories are more represented in the training dataset.

8 Future Plans

We plan on applying the models we have fine-tuned on the other 99 percent of unaccepted petitions. Since the unaccepted petitions do not have ground truth labels, a future validation method to validate the accuracy of the models on the unaccepted petitions will have to be developed. A method could be hand labeling some of the unaccepted writs to be used as training labels.

9 Contributions

The following contributions were made by each member of the team:

1. Haikal Rozaidi: Scraped and organized full-text petitions and labels from the datasets. The set of petitions were also preprocessed before being run on the models.

2. Trey Dixit: Mainly responsible in writing and running the embeddings code as well as the main person involved in running the legalBERT part of the project.

3. Dylan Yang: The main person to run the neoBERT part of the project.

4. Abhinay Puttay: The main person to run the regular BERT model on the full-text set of petitions.

All members assisted in writing the final report as well as the presentation slides. All members were involved in presenting the paper in class during presentation week.

10 Outside Acknowledgements

The American Institutions and Methodology Lab was instrumental in helping scrape the proquest dataset. Scraping supremecourt.gov was done by the CSE 5525 research group, however.

11 Appendix

11.1 Issue Area Key

1:Criminal Procedure
2:Civil Rights
3:First Amendment
4:Due Process
5:Privacy
6:Attorneys
7:Unions
8:Economic Activity
9:Judicial Power
10:Federalism
11:Interstate Relations
12:Federal Taxation
13:Miscellaneous
14:Private Action

11.2 Case Disposition Key

1: Stay, petition, or motion granted
2: Affirmed (including modified)
3: Reversed
4: Reversed and remanded
5: Vacated and remanded
6: Affirmed and reversed (or vacated) in part
7: Affirmed and reversed (or vacated) in part and remanded
8: Vacated
9: Petition denied or appeal dismissed
10: Certification to or from a lower court

12 References

Jones, Bryan D., Frank R. Baumgartner, Sean M. Theriault, Derek A. Epp, Cheyenne Lee, Miranda E. Sullivan. 2023. Policy Agendas Project: Codebook.

Harold J. Spaeth, Lee Epstein, et al. 2024 Supreme Court Database, Version 2024 Release 1. URL: http://supremecourtdatabase.org

Le Breton, L., Fournier, Q., El Mezouar, M., & Chandar, S. (2025). NeoBERT: A Next-Generation BERT [Preprint]. arXiv. URL: https://arxiv.org/abs/2502.19587

Supreme Court of the United States. 2024. Supreme Court of the United States Official Website. URL: https://www.supremecourt.gov.

ProQuest. 2024. ProQuest Research Database. URL: https://www.proquest.com.

Choi, Jonathan H., How to Use Large Language Models for Empirical Legal Research (August 9, 2023). Journal of Institutional and Theoretical Economics (Forthcoming), Minnesota Legal Studies Research Paper No. 23-23, Available at SSRN: https://ssrn.com/abstract=4536852

Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., & Androutsopoulos, I. (2020). LEGAL-BERT: The muppets straight out of law school. *arXiv preprint arXiv:2010.02559*.