LegalAIze: A Text Prediction Model for the Legal Domain

Divyansh Verma, Shiven Barbare, Urvi Midha, Vamsi Varma Kalidindi

College of Computing
Georgia Institute of Technology

Team ID: O

1 Introduction

Text prediction and autocomplete systems have become integral to modern applications, ranging from email clients to search engines. These systems rely on Natural Language Processing (NLP) models trained on vast amounts of textual data to predict the next word or sequence of words, thereby enhancing user experience and efficiency. While there has been significant progress in developing general-purpose text prediction models, their effectiveness in specialized domains, such as legal language remains limited due to the unique jargon, complex sentence structures, and specific legal terminology.

The legal domain presents a unique opportunity for NLP due to the highly formal and specialized language used in legal documents (Zhong et al., 2020). Generalized large language models are too general and lack the systematic language expected from legal text, making them unusable for domain specific applications. Our findings indicate that the previous guidelines for pre-training and fine-tuning, often blindly followed, do not always generalize well in the legal domain (Chalkidis et al., 2020).

Through this project we aim to address this gap by fine-tuning a general pretrained language model on a legal dataset, with the goal of creating a robust text prediction and autocomplete system tailored to the legal domain. The main idea that this project aims to tackle is to experiment and fine-tune existing general purpose language models (particularly GPT-2 and BERT) and build a tool that specializes in the legal domain for text prediction and auto completion purposes.

Within the legal domain, some applications such as Machine Summarization and Text Generation have seen little activity compared to the more popular fields of Classification and Information Extraction. Given the current interest in general text generation through popular exposure to applications such as ChatGPT, we expect a marked increase in the volume of Text Generation papers in the legal domain in the decade to come (Katz et al., 2023). Our approach will look to bridge the gaps and bring higher performance of models like those of generalized LLM's specifically to the legal domain. This can be achieved by using intermediate training strategies to enhance pre-trained language models' performance in the text auto-completion task and quickly adapt them to specific domains (Lee et al., 2021).

2 Related Work

The paper (Qu et al., 2020) describes a methodology for text generation in the Chinese language using pre-trained GPT-2 and BERT. Both models are trained using new corpora and used to predict entire sentences from a starting word/token. The study highlights the differences between GPT-2 and BERT in terms of their working, advantages, and use cases for text generation (GPT-2 for sentence generation, and BERT for word prediction). The paper also outlines an intuitive web page as a test-bed for evaluating the text generation system. The study directly relates to our approach of using pre-trained LLMs and training them using domain-specific datasets. While it deviates from our objectives in its Chinese-language focus and granularity of text generation (predicting sentences and not words), the study still provides relevant insights into using transformer-based models for domain specific word prediction systems.

The paper (Aliprandi et al., 2008) introduces a novel word prediction system, FastType that combines Part-of-Speech (POS) Tagging and n-gram Models for the Italian language. The algorithm uses a linear combination of probabilities of POS trigrams and Tagged Word bigrams. FastType is highly suited to languages with relatively high degrees of inflection (such as Italian, French, Spanish). Another highlight of the paper is its unique user-interface that evaluates the model with metrics like Keystrokes Saving and Keystrokes until Com-

pletion. English is a relatively simpler language as far as inflection, vocabulary and contextual ambiguity are concerned, so FastType can prove to be efficient for text generation with the language. The methodology in the paper is not directly relevant to the scope of our project as we aim to use pre-trained transformers as a starting point, nevertheless, it outlines an innovative approach to text generation.

The study in (Lee et al., 2021) tries to overcome the challenges faced by deep neural networks to perform domain specific text auto complete tasks. The paper proposes to fine tune existing LLM architectures using a novel Next Phrase Prediction method (NPP) to complete a partial query with adequate phrases. NPP involves two major steps: Phrase Extraction and Generative Question Answering. Phrases are extracted using constituency parsing to retrieve qualitative phrases and are grouped into sets according to their types. After retrieving the phrases the language model is trained to predict the correct next phrase in a generative QA task setting. The study focuses on email and academic text completion domain which is different from our legal domain but we find the learnings useful with meaningful knowledge transfer across domains.

The paper (Al-Mubaid, 2003) studies text prediction using an Ltest learning model leveraging lexical and syntactic information about words. Words are encoded keeping their surrounding context and relationships with other occurrences in the corpus. This style of encoding then enables data mining tools to generate sets of logic formulae which is in turn used by word predictor models to determine the next word provided the context. Experimentation was done on Wall Street Journal dataset which is a different domain than our legal domain but the methodology transfers well across different domains as the legal domain also uses very strict rules with respect to how documents are written just like how WSJ articles stick to a specific lexica for their articles.

The study in (Zhong et al., 2020) provides a detailed overview of Legal AI. It covers two main approaches - symbol-based methods that rely on interpretable hand-crafted rules, and embedding-based methods that utilize neural networks and learned representations. It highlights three main challenges: knowledge modeling, legal reasoning, and interpretability. While these are crucial, the paper doesn't fully explore how these challenges interact or how addressing one might impact the others. It also covers key applications including Legal Judgment Prediction, Similar Case Matching, and Legal Question Answering, highlighting current limitations such as the performance gap between AI models and human experts. The paper suggests future directions, including focusing on interpretability and fairness. Overall, the paper provides a solid overview of the current state and future of Legal AI, emphasizing the need for further research to address challenges and ethical issues. Our project will aim to fill some of these gaps, particularly in domain-specific language modeling and knowledge integration.

The paper (Ambulgekar et al., 2021) on "Next Word Prediction Using Recurrent Neural Networks" explores how Recurrent Neural Networks (RNNs), especially Long Short-Term Memory (LSTM) networks, can be used for predicting the next word in a text. The method focuses on predicting the next word by processing the text letter by letter, after a user has typed 40 characters. LSTM networks are used because they can remember long-term patterns in text. While this letter-by-letter prediction method might work well for general language tasks, it is less effective for specialized fields like legal text prediction, where predicting full words or phrases is crucial to maintain accuracy and flow in formal documents. The model was trained on a corpus of Nietzsche's writings and aims to improve prediction speed and accuracy compared to traditional models like Decision Trees and SVM. While the model architecture is scalable, it could be further improved with more or different datasets.

The paper (Chalkidis et al., 2020) explores different strategies for adapting BERT to the legal domain and presents LEGAL-BERT, which utilize BERT for various NLP tasks. The authors investigate three approaches: using pre-trained BERT out-of-the-box, further pre-training BERT on legal corpora, and pre-training BERT from scratch on legal data. Key findings include that domain-specific pre-training outperforms generic BERT, and that an expanded hyperparameter search during fine-tuning significantly impacts performance. While this work demonstrates the benefits of domain adaptation for BERT in legal text, it focuses primarily on classification tasks rather than text prediction. Our project hopes to utilize some of these techniques to achieve high accuracy specifically for the purpose of text prediction and generation and as a comparison metric.

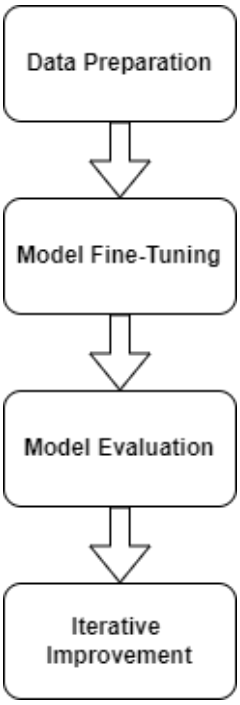

Figure 1: High-Level Project Flow

The paper (Leitner et al., 2019) presents an approach for Named Entity Recognition (NER) in German legal documents. The authors developed a dataset of German court decisions annotated with semantic classes, which were in turn mapped to coarse-grained classes. They applied Conditional Random Fields (CRFs) and bidirectional Long Short-Term Memory networks (BiLSTMs) to this dataset for NER. Key findings include that BiLSTM models outperformed CRFs and that legal named entities differ significantly from those in news texts, with legal norms and case references being more prevalent than typical person/organization entities. While this work demonstrates the effectiveness of neural NER models on legal texts, it is limited to German documents and may not generalize to the English language which our project is focused on. This paper gives an insight into some techniques that we can use for named entity recognition and how the legal domain performs compared to generic text data.

3 Proposed Methodology

The project utilizes general pre-trained language models as the starting point. We then fine-tune these models using a legal dataset that includes a variety of legal documents, such as case law, statutes, contracts, and legal briefs.

The project flow begins with cleaning and preprocessing the legal dataset to ensure that it is in a suitable format for model training. The dataset that we used for this purpose is called Pile-of-Law. This is followed by adapting the pre-trained model to the legal domain by training it on the prepared legal dataset. This step involves adjusting the model's parameters to better capture legal language patterns and terminology. The final steps involve evaluating the model's performance on legal text prediction and autocomplete tasks, using accuracy metrics and relevance of predictions. Based on the evaluation results, we iteratively refine the model to improve its performance.

The decision of fine-tuning GPT-2 and BERT, specifically was based on a survey of different LLMs and their usefulness for the outcomes of this project. Language Models like XLNet, LLaMA and T5 were not considered because of their high complexity and requirement of significantly more computational resources.

4 Dataset

We sourced legal texts from the Pile-of-Law dataset, which is substantial at 291.5 GB and contains over 45 data instances like 'us_bills', 'tos', 'un_debates' etc. For our study we chose to use the 'us_bills' instance as our main data source mainly due to relevancy and compute limitations.

Each instance of data has 4 features:

- **'created_timestamp'**: When the document was created.
- **'downloaded_timestamp'**: When the document was scraped.
- **'url'**: The source url
- **'text'**: The document text. We use this field for our task.

5 Experiments and Results

In this section, we present the experimental setup, model fine-tuning procedures, and the results obtained for two pre-trained transformer models, GPT-2 and BERT. The performance of the fine-tuned models was compared with the respective baseline pre-trained models for the same next word prediction task.

5.1 Preprocessing

In our preprocessing pipeline, we implemented essential tasks for preparing legal documents for BERT and GPT-2 fine-tuning. These include removal of tab and new line characters. Another common feature of the documents was a recurrence

of roman numerals in brackets, which were also removed to have cleaner data. The other parts of the data were already clean and we did not choose to add stop-word removal, since all the words in context of a legal document would be useful in the process of training.

5.2 Experiments

During training, we employed a custom chunking process to handle long US bills, ensuring that each input sequence stayed within the token limit of 512. If a bill exceeded this token limit, it was split into smaller, consecutive chunks of 512 tokens. For example, a bill with 1500 tokens would be divided into three chunks: one for tokens 1–512, another for 513–1024, and the last for 1025–1500. This method ensured that the entire bill was utilized during training, even when it required multiple chunks. Each chunk was processed independently by the model, allowing it to learn from the entire content of the bill while maintaining the model's token size constraints.

During validation, predictions were made on sentences of variable length. This was done to monitor performance of model on a varied spectrum of input text lengths. We wanted to ensure that model performance was acceptable even if the input text was small (use case: user just started writing the document) as well as maintain viability when the input text was larger (use case: user is more than halfway done writing their document). This was done by implementing a custom input text generator logic that generates the first text of 15 tokens and then appending 20-40 tokens randomly to increase the input text length and making predictions for each input text of variable length thus created.

Accuracy was used as the performance metric.

Accuracy measures the model's success at predicting the correct next token in sequences from the test data. The accuracy is calculated as mentioned below:

$$\text{Accuracy } (A) = \frac{\#\text{ of correct word predictions}}{\#\text{ of testing samples}}$$

In addition to overall accuracy, top-k accuracy was also computed. Top-k accuracy measures how often the correct word appears within the top-k predicted words (where k=3). This metric is useful in assessing the model's ability to produce relevant candidates, even if it doesn't always make the top prediction. Top-k accuracy is calculated as:

$$\text{Top k Accuracy } (A) = \frac{\#\text{ of times the correct word is in the top k predictions}}{\#\text{ of testing samples}}$$

These two metrics were used to evaluate the effectiveness of the models in predicting the next word in sequences and comparing the improvements achieved by fine-tuning.

5.3 Results

The results of the experiments, comparing the pre-trained models with their fine-tuned versions, are summarized in the table below:

Model	Accuracy (%)	Top-3 Accuracy (%)
GPT-2 (Pre-trained)	50.92	65.57
GPT-2 (Fine-tuned)	59.71	73.99
BERT (Pre-trained)	32.31	33.49
BERT (Fine-tuned)	35.26	35.36

Table 1: Performance of GPT-2 and BERT models before and after fine-tuning.

For GPT-2, fine-tuning resulted in a significant improvement in both overall accuracy and top-3 accuracy. The overall accuracy increased from 50.92% to 59.71%, and top-3 accuracy rose from 65.57% to 73.99%. This demonstrates that fine-tuning enabled the model to better capture the relevant patterns in the bill text, leading to more accurate predictions of the next word. We also observe that the top-3 accuracy is much higher than the accuracy in case of GPT-2.

In contrast, for BERT, the improvements from fine-tuning were modest. The overall accuracy increased slightly from 32.31% to 35.26%. A similar improvement was observed in the top-3 accuracy as well.

To demonstrate the model's predictions, the figure 2 provides a snapshot of its performance while processing one of the bills. The table captures key details, including the input text length, the actual next word, the predicted next words (both top-1 and top-3), and whether the predictions matched the ground truth.

6 Baseline Comparison

We used LegalBERT as the SOTA/baseline to measure the effectiveness of our approach. LegalBERT is a specialized version of BERT that is specifically fine-tuned on legal texts. So, it is basically a

```
Processing bill 11 of length 3764 characters...
```

	bill_number	input_text_length	actual_next_word	predicted_next_word	top_k_word_prediction	top_word_prediction
0	11	15	u	[u, of,]	True	True
1	11	55	public	[public, interest, improper]	True	True
2	11	78	the	[the, congress,]	True	True
3	11	98	the	[the, , a]	True	True
4	11	126	child	[child, payments, federal]	True	True
5	11	153	collection	[collection, enforcement,]	True	True
6	11	180	re	[re, ri, ab]	True	True

Figure 2: Model Predictions

smaller version of BERT that is specialized for all things legal.

Using the same approach as GPT-2 and BERT, we perform next-word prediction with LegalBERT and obtain the following metrics:

- Accuracy: 38.28%
- Top-3 Accuracy: 38.44%

7 Conclusion

We were able to run pre-trained models like BERT and GPT-2 on a subset of US Bills data to find the accuracy of next-word prediction tasks. We also fine-tuned these models using the US Bills text and calculated the accuracy using the same validation set. Additionally, we calculated the top-k accuracy as well for the pre-trained and fine-tuned models. We observe that GPT-2 performs much better than BERT in all cases, which is to be expected, since GPT-2 is designed for generation tasks while BERT is more aligned to solve for Masked Language Modeling (MLM) tasks. The top-k accuracy method yielded better results for GPT-2, while it did not improve the results of the BERT models by much.

We then ran the same experiments on LegalBERT which is an existing model built using a large corpus of legal text. This model only performs slightly better than the pre-trained and fine-tuned BERT models. The top-k accuracy does not give any significantly better results.

8 Limitations

The biggest limitation for this project was a lack of compute resources for training models. The model efficiency can be improved by increasing the size of the training data, i.e., training on more bills and incorporating even more data aspects like Terms of Service, UN debates etc from the dataset Pile-of-Law.

Due to this problem statement's fairly niche application there is still a lack of reference resources and research work. We hope this project will be good reference for future research work.

Defining a term as a 'legal' term is fairly difficult. What makes a text suitable for the legal domain often has a specific combination of semantic, grammar and vocabulary context. Developing a novel method for the above would require a high degree of expertise in the domain and was out of the scope for this project. We were limited to general text evaluation metrics but further research work can be done in order to validate and rate text prediction in a legal specific context.

References

Hisham Al-Mubaid. 2003. Context-based word prediction and classification. In *Proceedings of the 18th International Conference on Computers and their Applications CATA*, volume 2003, pages 384–388. Citeseer.

Carlo Aliprandi, Nicola Carmignani, Nedjma Deha, Paolo Mancarella, and Michele Rubino. 2008. Advances in nlp applied to word prediction. *University of Pisa, Italy February*.

Sourabh Ambulgekar, Sanket Malewadikar, Raju Garande, and Bharti Joshi. 2021. Next words prediction using recurrent neuralnetworks. In *ITM Web of Conferences*, volume 40, page 03034. EDP Sciences.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. Legal-bert: The muppets straight out of law school. *arXiv preprint arXiv:2010.02559*.

Daniel Martin Katz, Dirk Hartung, Lauritz Gerlach, Abhik Jana, and Michael J Bommarito II. 2023. Natural language processing in the legal domain. *arXiv preprint arXiv:2302.12039*.

Dong-Ho Lee, Zhiqiang Hu, and Roy Ka-Wei Lee. 2021. Improving text auto-completion with next phrase prediction. *arXiv preprint arXiv:2109.07067*.

Elena Leitner, Georg Rehm, and Julian Moreno-Schneider. 2019. Fine-grained named entity recognition in legal documents. In *International conference on semantic systems*, pages 272–287. Springer.

Yuanbin Qu, Peihan Liu, Wei Song, Lizhen Liu, and Miaomiao Cheng. 2020. A text generation and prediction system: Pre-training on new corpora using bert and gpt-2. In *2020 IEEE 10th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, pages 323–326.

Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. How does nlp benefit legal system: A summary of legal artificial intelligence. *arXiv preprint arXiv:2004.12158*.