CourtlyCrafter: A Contextual Legal Language Model for Automated Text Generation

Kaif Asif Shaikh
D. Y. Patil International University
Pune, India

***Abstract*—The ability to automatically generate coherent, relevant text has numerous applications in the legal domain for drafting documents and analyzing case law. We present CourtlyCrafter, an open-sourced contextual text generation language model specialized for legal and court documents based on BERT architecture. CourtlyCrafter is pretrained on a large-scale multinational corpus of over 18 billion legal tokens encompassing legislation, court cases, and contracts. We detail the web-scraped dataset curation, preprocessing, model training and sampling methodology. Evaluations show our legal language model exceeds baseline generic BERT models in text perplexity and human rating metrics for coherence, fluency and domain relevance. Text samples demonstrate capable zero-shot generalization with accurate terminology. Live online demos highlight promising real-world use for automated report writing and as legal research assistants. CourtlyCrafter provides a strong pretrained foundation for conditional legal text generation and style transfer tasks. We discuss current limitations and future enhancements integrating structured semantic knowledge into the model.**

I. Introduction

Text generation has seen rapid progress with deep contextual language models like GPT-3 demonstrating human-like fluency and coherence in generated text. However, for many domain-specific applications like the legal field, generic models do not capture the nuances of terminology, writing conventions and reasoning required. Pretraining models on in-domain text corpora provides specialization for both fluency and relevance.

There are several use-cases for automated legal text generators like assisting drafting of legislative documents, summarizing lengthy court judgments or compiling applicable legal sections for a case. Accuracy of terminology and citation is critical in these tasks. Current solutions rely on templating and simple rule-based methods with little ability to adapt.

Recent techniques in domain-adaptive pretraining now enable learning contextual generation models specialized for niche domains like law. Ensuring careful dataset curation and comparing domain-specific models to generic baselines is important to quantify value-add.

II. LexFiles: English Multinational Legal Corpora

The LexFiles dataset represents an extensive collection of English Multinational Legal Corpora. It encompasses diverse legal documents from various jurisdictions and serves as a valuable resource for legal language understanding and modeling. This dataset comprises several subsets, each catering to distinct legal domains and regions, such as EU Legislation, EU Court Cases, UK Legislation, UK Court Cases, Indian Court Cases, US Contracts, US Court Cases, US Legislation, Canadian Legislation, and Canadian Court Cases.

A. Builder Configurations

The dataset is structured using a LexFilesConfig class, allowing for different configurations based on the legal corpus being accessed. Each configuration contains a unique name, description, version, and URL pointing to the specific dataset subset.

B. Dataset Information

The dataset offers valuable insights into legal language through its features, particularly the "text" attribute, which holds the textual content of legal documents. With no supervised keys, the LexFiles dataset is suitable for various natural language processing (NLP) tasks, including language modeling, text generation, classification, and information extraction within the legal domain.

III. Data Retrieval and Usage

The dataset can be accessed via Hugging Face Datasets API using the provided URLs for each subset. Upon download and extraction, it offers access to subsets in three splits: training, validation, and testing. These subsets contain JSONL files with examples that can be used for various NLP tasks or fine-tuning language models specific to legal contexts.

The LexFiles dataset proves to be a crucial resource for NLP practitioners, researchers, and legal professionals aiming to analyze, model, or develop language-based applications within legal domains across different jurisdictions.

IV. Training and Dataset Preprocessing

A. Environment Setup and Libraries

The project utilizes various Python libraries such as *comet_ml*, *pandas, datasets,* and *transformers.* Additionally, the environment setup involves ignoring warnings and initializing *CometML* for project tracking.

B. Data Loading and Merging

The training data is loaded using pandas from CSV files *(train.csv, val.csv, test.csv)*. The datasets are then concatenated and reset for a unified representation using *pd.concat()* and *reset_index().*

C. Dataset Creation and Splitting

The merged *dataset* is converted to a *Dataset* object from the datasets library. Further, the dataset is split into training and testing subsets, with 80% of the data allocated for training and 20% for testing using *train_test_split().*

D. Tokenization and Preprocessing

The dataset is tokenized using the *AutoTokenizer* from the *transformers* library. A *preprocess_function* is defined to tokenize text examples, applying padding, truncation, and limiting the sequence length to 256 tokens.

E. Model Configuration and Preparation

The project utilizes the *AutoModelForCausalLM* for language modeling, instantiated from the *prajjwal1/bert-mini* pre-trained model. Additionally, the *DataCollator* for *LanguageModeling* is set up with tokenizer configurations and *mlm* parameter.

F. Metric Computation and Experiment Logging

A *compute_metrics* function calculates evaluation metrics such as accuracy, precision, recall, and F1-score using *precision_recall_fscore_support* and *accuracy_score*. The function additionally logs metrics and confusion matrices to *CometML* during training epochs.

G. Model Training

Training settings and arguments including output directory, evaluation strategy, learning rate, epochs, weight decay, and reporting to *CometML* are configured via *TrainingArguments*. The model is then trained using the Trainer object, passing in the defined model, arguments, dataset, metrics function, and data collator.

V. Training and Dataset Preprocessing

A. Training Process Overview

This subsection provides an overview and analysis of the model's training process. It discusses the training duration, convergence patterns, and any observed anomalies or trends during the epochs.

B. Performance Metrics

Metrics such as accuracy, precision, recall, and F1-score are presented and discussed in terms of model performance and effectiveness.

C. Error Analysis

I explored specific instances where the model struggled and provides insights into potential causes for these errors.

D. Model Interpretability

This subsection aims to interpret the model's decisions, highlighting which parts of the input text were significant in generating specific outputs. Methods such as attention visualization or saliency maps might be employed like Pooler Layers, Encoder and Decoder Layers, Multi-Head Attention Layers etc.

E. Generalization and Robustness

The discussion focuses on the model's generalization capability and robustness across various legal domains or specific legal contexts. I investigated the model's ability to adapt and perform adequately on unseen data by running model training and evaluation.

F. Ethical Considerations

This part addresses ethical considerations pertaining to the deployment and use of the model within the legal domain. It highlights potential biases, fairness, transparency, and accountability issues that arose during the training process.

G. Conclusion of Model Evaluation

Summarizing the findings from the model evaluation, this section presents conclusions regarding the model's performance, limitations, potential areas for improvement, and its readiness for deployment in legal settings.

Conclusion

In this project, we embarked on developing a Text Generation Legal Language Model (LLM) named "Courtly Crafter" aimed at enhancing text generation capabilities within the legal domain. The endeavor commenced with the acquisition and exploration of diverse English Multinational Legal Corpora, encompassing EU and UK legislation, court cases, US contracts, and more. Leveraging Exploratory Data Analysis (EDA), we gained valuable insights into token distributions, sentence structures, and word analytics, contributing to a comprehensive understanding of the dataset's characteristics.

Subsequently, the training phase involved the orchestration of data preparation, tokenization, and model training utilizing the "prajjwal1/bert-mini" transformer architecture. Evaluative metrics, including accuracy, precision, recall, and F1 score, were harnessed to assess the model's performance. Despite encountering certain challenges during the training phase, such as index errors during dataset indexing, the model was trained with a focus on bolstering its language generation abilities within the legal realm.

The project's outcomes and discussions revolve around the viability and potential of "Courtly Crafter" in generating coherent and contextually relevant legal text. Continuous optimization and enhancement remain integral for the model's refinement and its applicability in real-world legal language generation tasks.

References

[1] The HuggingFace Datasets Library: https://huggingface.co/datasets

[2] Prajjwal1/bert-mini Transformer Model: https://huggingface.co/prajjwal1/bert-mini

[3] Transformer Trainer Module Documentation: https://huggingface.co/transformers/main_classes/trainer.html

[4] Comet.ml for Experiment Tracking: https://www.comet.ml/