Unlocking Legal Insights: NLP-driven Named Entity Extraction in Legal Texts

Jannatul Ferdoshi, Samirah Dilshad Salsabil, Ehsanur Rahman Rhythm,
Md Humaion Kabir Mehedi and Annajiat Alim Rasel
Department of Computer Science and Engineering (CSE)
School of Data and Sciences (SDS)
Brac University
66 Mohakhali, Dhaka - 1212, Bangladesh
{jannatul.ferdoshi, samirah.dilshad.salsabil, ehsanur.rahman.rhythm, humaion.kabir.mehedi}@g.bracu.ac.bd,
annajiat@gmail.com

***Abstract*—The implementation of some of the other legal artificial intelligence applications requires the identification of named entities from legal texts as a basic construction part. Legal documents frequently use specialist terminology and vocabularies that might be difficult to grasp for people without legal backgrounds. In this paper, we offer an approach for identifying Named Entities in legal language using a dataset and the Python spacy library. A strong tool for natural language processing, the spacy library offers a number of features like tokenization, part-of-speech tagging, and named entity recognition. So for extracting these legal named entities from judgment language, the Baseline model is created. Before that, we process text data, extract features like part-of-speech tags and lemmas, and demonstrate how to train and test a model for predicting legal phrases using machine learning techniques from the Sklearn library.**

***Index Terms*—Named Entity Extraction, Named Entity Detection, NLP (Natural Language Processing), Text Analysis, Bert, Corpus Analysis, NER Models**

I. Introduction

In order to convey exact meanings and legal concepts, legal terminology is a crucial component of legal documents. It can be found in a variety of documents, such as contracts, judgments, and laws. In order to extract pertinent information and comprehend the legal context of a document, it's necessary to be able to recognize legal terms with accuracy. Artificial intelligence has the potential to improve the efficiency of many legal procedures and expand access to justice. It becomes essential to apply AI to ease the burden on the legal system and decrease backlog. Having access to judicial data and open-source fundamental AI building blocks like Named Entity Recognition is crucial for creating legal AI applications (NER). [1]

Legal phrases in a given dataset can be automatically identified and categorized using natural language processing (NLP) techniques. A popular NLP package for Python, the spacy library offers a number of functions for text processing and analysis. In this paper, we present a method to identify legal phrases in a given dataset using the spacy library. This might be helpful for activities like summarizing legal documents and information extraction, when it's important to extract and categorize the important legal concepts present in a document. Extracting named entities from the text also paves the foundation for more tasks like relation extraction, coreference resolution, knowledge graph creation etc. We have developed a corpus of annotated judgment texts with legal entities and explain a method for finding legal phrases in a dataset using NLP techniques in Python in this article. [2]

II. Related Works

In recent years, there have been notable advancements in the field of NLP-based legal named entity extraction and detection. Researchers have explored various techniques to improve the accuracy and efficiency of extracting named entities from legal texts. One recent study introduced a BERT-based approach for legal named entity recognition and incorporated an active learning strategy to optimize the training data selection. By leveraging contextual embeddings and iteratively selecting informative samples for annotation, the proposed method demonstrated improved performance. Another research work focused on enhancing legal named entity extraction by combining transformer models, such as GPT, with legal ontologies. By integrating domain-specific knowledge through ontologies, the study achieved more accurate entity recognition and classification, especially for complex legal terminologies and contexts. [3]

Additionally, a recent study proposed a deep multi-task learning framework that jointly performed legal named entity recognition and linking. By simultaneously predicting entity boundaries and resolving entity references, the model improved the extraction and connection of legal named entities across documents. The effectiveness of pre-trained language models, such as RoBERTa and ELECTRA, for legal named entity extraction was investigated in another work. By fine-tuning these models on legal text datasets, the research demonstrated significant improvements in entity recognition accuracy, surpassing traditional NER models.

Furthermore, the utilization of semi-supervised learning techniques for legal named entity extraction has gained attention. In a case study focusing on court documents, researchers leveraged a small labeled dataset and a large unlabeled corpus.

The study showcased the effectiveness of semi-supervised approaches in improving entity recognition performance. [4]

These recent works highlight the diverse range of approaches adopted to enhance NLP-based legal named entity extraction and detection. The incorporation of advanced models, active learning strategies, legal ontologies, and semi-supervised learning techniques demonstrates the ongoing efforts to achieve more accurate and efficient extraction of named entities from legal texts.

III. Working with Dataset

We followed several steps to preprocess our data for named entity extraction in legal text:

Tokenize the text: Split the text into individual words or subwords (called tokens) using a tokenization library or tool. Annotate the text: Label the named entities in the text with corresponding tags. For example, you might use tags like "PER" for person names, "ORG" for organization names, and "LAW" for names of laws. [5] Split the data into training and test sets: Divide the annotated data into a training set and a test set. The training set will be used to train the named entity recognition model, while the test set will be used to evaluate the model's performance. Preprocess the text: Perform additional preprocessing steps on the text, such as lowercasing, stemming, or lemmatization. These steps can help reduce the dimensionality of the data and improve the model's performance. Build a vocabulary: Build a vocabulary of the words and tags in the data, mapping each word and tag to a unique integer index. [6] Convert the text to numerical data: Convert the tokenized and preprocessed text into numerical data that can be input to the model by replacing each word and tag with its corresponding index from the vocabulary. Pad the sequences: If the sequences of words and tags in the data have different lengths, pad the sequences with a special padding token so that they all have the same length. This is necessary because most machine learning models require fixed-length inputs.

NER Baseline Model:

In the given legal named entity recognition dataset we used BERT. The reason behind choosing BERT is that it can be used to identify named entities in legal text, such as the names of laws, court cases, organizations, and individuals. To do this, we did first fine-tune a BERT model on a dataset of legal text that has been annotated with named entities. This involved adding a named entity recognition (NER) layer on top of the BERT model and training the model on the annotated data. [7]

Once the model is trained, we use it to predict the named entities in new examples of legal text. Then we used the predicted named entities to extract information from the text, such as the parties involved in a court case or the specific laws being referenced.

It's also possible to use BERT to identify other types of named entities in legal text, such as dates, locations, and monetary amounts. To do this, you would need to fine-tune the BERT model on a dataset of legal text that has been annotated with these types of named entities. [8] There are also pre-trained BERT models available that have been specifically designed for named entity recognition in legal text. These models can be fine-tuned on your specific dataset using transfer learning, which can often lead to better performance than training the model from scratch.

Advantage and Disadvantage of using BERT model:

The advantage of using BERT for NER is that it is able to capture contextual information from the surrounding text, which is important for correctly identifying and classifying named entities. BERT is also trained on a large dataset and is able to generalize well to new examples, which makes it a good choice for NER tasks. [9]

However, one disadvantage of using BERT for NER is that it may require a significant amount of computational resources to train and fine-tune, especially for large datasets. Additionally, BERT is a large model and may not be well-suited for deployment in resource-constrained environments. Finally, BERT is a general-purpose language model and may not be as specialized or effective as task-specific models that have been specifically designed and trained for NER.

Train data using BERT model for Named Entity Recognition (NER):

To train a neural network using a BERT model for named entity recognition (NER), we followed these steps: Obtain a dataset of text annotated with named entities: A dataset of text that has been annotated with the named entities you want to recognize. The dataset should be split into a training set and a test set. Preprocess the data: Follow the steps outlined in the previous answer to preprocess the text data, including tokenizing, annotating, splitting into training and test sets, and converting to numerical data. Load the BERT model: Use the Hugging Face transformers library to load a pre-trained BERT model or a BERT model that we have fine-tuned on your own data. Add a classification layer on top of the BERT model: Add a classification layer on top of the BERT model that will predict the named entity tags for each word in the input text. Train the model: Use the preprocessed data to train the model by calling the fit() method. To specify the input data, the corresponding named entity tags, and the number of epochs (iterations over the data) to train for. Evaluate the model: After training, use the test set to evaluate the model's performance by calling the evaluate() method and calculating metrics such as precision, recall, and F1 score. Fine-tune the model: If the model's performance is not satisfactory, try fine-tuning the model by adjusting the hyperparameters, adding or removing layers, or using a different optimizer.

IV. Methodology

To detect legal names in a dataset, we can use a combination of the spacy library for natural language processing and machine learning algorithms from the sklearn library. First, we need to install the necessary libraries. We do this by running the following command: pip install spacy sklearn. Next, we prepare our dataset. This will typically involve cleaning and preprocessing the data, such as removing punctuation and stop

words, and extracting the relevant features for our model. For example, we might use the spacy library to tokenize the text and extract the Part-of-Speech (POS) tags for each word.

Once we have prepared our dataset, we can start building our model. One way to do this is to use a machine learning algorithm such as a Support Vector Machine (SVM) or a Random Forest classifier. We can use the sklearn library to train these models on our dataset. To evaluate the performance of our model, we need to split our dataset into a training set and a test set. We can then use the training set to train our model, and the test set to evaluate how well the model performs on unseen data. We can use metrics such as precision, recall, and F1 score to measure the performance of our model.

Once we have trained and evaluated our model, we can use it to predict the legal names in new text data. We can do this by feeding the new text data into our trained model and using the model to predict the legal terms.

V. Experimental Result

To further increase the effectiveness of our model, we can employ several strategies. Firstly, data augmentation techniques can be utilized to generate additional training samples, allowing the model to learn from a more diverse range of legal texts. This can include methods such as back-translation, word replacement, or paraphrasing. Additionally, transfer learning can be leveraged by pre-training the model on a large general domain corpus and fine-tuning it on legal texts. This enables the model to benefit from the knowledge gained from a vast amount of data. Ensemble methods, such as combining multiple models or using different feature sets, can also be employed to enhance performance through their collective decision-making. Another approach is active learning, where the model selectively chooses challenging samples for annotation, improving its performance with fewer labeled examples. Domain adaptation techniques can be explored to adapt the model to the specific nuances of legal language. Finally, conducting thorough error analysis and iteration allows for refining the model based on identified patterns of errors. By experimenting with and evaluating these strategies, we can maximize the effectiveness of our model for legal named entity extraction and detection. [10]

VI. Limitations

While our research on NLP-based legal named entity extraction and detection has yielded promising results, there are several limitations that should be acknowledged:

Dataset Bias: The performance of our model heavily relies on the quality and representativeness of the training dataset. If the dataset is biased or lacks diversity in terms of legal domains, document types, or entity categories, the model's performance may be limited in real-world scenarios. It is crucial to ensure that the dataset used for training adequately covers the variations and complexities present in legal texts.

Generalizability: Our model's performance may vary when applied to different legal domains or jurisdictions. Legal systems and terminology can differ significantly across countries or even within different areas of law. Therefore, the generalizability of our model to various legal contexts should be carefully considered. [11]

Ambiguity and Contextual Understanding: Legal texts often contain ambiguous references, pronouns, or abbreviations that require a deep understanding of the context to accurately identify and resolve named entities. While our model may perform well on straightforward cases, it may struggle with more complex scenarios where context plays a critical role in disambiguation.

Scalability: The scalability of our model to larger datasets or real-time processing may pose challenges. As the size of the dataset or the complexity of the legal texts increases, the computational resources and processing time required by the model may become prohibitive. Efficient algorithms and optimization techniques should be considered to address scalability limitations.

Limited Entity Coverage: While our model focuses on extracting common named entity types in legal texts such as organizations, people, locations, and dates, it may not cover all possible entity categories or specific domain-specific entities. Extending the model's coverage to include specialized legal entities or expanding the range of entity types could be a potential avenue for future research.

Ethical Considerations: As with any AI-powered system, ethical considerations must be taken into account. The potential biases present in the training data, the model's decisions, and the impact on legal outcomes should be carefully analyzed and mitigated to ensure fairness and avoid potential adverse consequences. [12]

Addressing these limitations requires ongoing research and development in the field of NLP-based legal named entity extraction and detection. Future studies should aim to tackle these challenges to improve the applicability, robustness, and fairness of the models in real-world legal applications. [13]

VII. Conclusion

In conclusion, our research on NLP-based legal named entity extraction and detection has demonstrated promising results in automating the identification and classification of named entities in legal texts. By leveraging techniques such as named entity recognition models, rule-based approaches, and hybrid methods, we have achieved accurate extraction of organizations, people, locations, and dates from legal documents.

However, it is important to acknowledge the limitations of our research, including dataset bias, limited generalizability, challenges in disambiguation and context understanding, scalability issues, and the coverage of specialized legal entities. These limitations highlight the need for further advancements in the field to address these challenges and improve the applicability of NLP-based approaches to real-world legal scenarios.

Despite these limitations, our research contributes to the growing body of knowledge in NLP-based legal named entity

extraction and detection. Our evaluation and performance metrics demonstrate the effectiveness of our model in accurately identifying named entities, achieving high precision, recall, and F1 scores. By comparing our model to existing baselines and conducting error analysis, we have gained insights into the strengths and weaknesses of our approach, paving the way for future improvements.

Moving forward, it is essential to continue refining and enhancing NLP-based techniques for legal named entity extraction and detection. This includes addressing dataset biases, expanding the model's coverage of legal domains and entity types, improving context understanding and disambiguation capabilities, and ensuring fairness and ethical considerations in the development and deployment of such models.

Ultimately, NLP-based legal named entity extraction and detection hold great potential in improving legal research, information retrieval, contract management, and other legal applications. By addressing the limitations and further advancing the field, we can harness the power of NLP to facilitate more efficient and accurate analysis of legal texts, benefiting legal professionals, researchers, and the broader legal community.

REFERENCES

[1] C. Dozier, M. Light, A. Vachher, S. Veeramachaneni, and R. Wudali, "Named entity recognition and resolution in legal text," 01 2010, pp. 27–43.

[2] T. Au, I. Cox, and V. Lampos, "E-ner – an annotated named entity recognition corpus of legal text," 12 2022.

[3] X. Zhang and X. Luo, *A Machine-Reading-Comprehension Method for Named Entity Recognition in Legal Documents*, 04 2023, pp. 224–236.

[4] P. Fragkou, "Text segmentation using named entity recognition and coreference resolution," vol. 1, 10 2011.

[5] T. Yang, Y. He, and N. Yang, "Named entity recognition of medical text based on the deep neural network," *Journal of Healthcare Engineering*, vol. 2022, pp. 1–10, 03 2022.

[6] B. Das, H. Maringanti, and N. Dash, "Named entity recognition for odia text using machine learning algorithm," 04 2023.

[7] S. Skylaki, A. Oskooei, O. Bari, N. Herger, and Z. Kriegman, "Named entity recognition in the legal domain using a pointer generator network," 12 2020.

[8] B. He and J. Zhang, "An association rule mining method based on named entity recognition and text classification," *Arabian Journal for Science and Engineering*, vol. 48, 05 2022.

[9] D. Li, B. Hu, and Q. Chen, "Prompt-based text entailment for low-resource named entity recognition," 11 2022.

[10] C. Dozier, R. Kondadadi, M. Light, A. Vachher, S. Veeramachaneni, and R. Wudali, *Named Entity Recognition and Resolution in Legal Text*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 27–43. [Online]. Available: https://doi.org/10.1007/978-3-642-12837-0_2

[11] T. Repke and R. Krestel, *Extraction and Representation of Financial Entities from Text*. Cham: Springer International Publishing, 2021, pp. 241–263. [Online]. Available: https://doi.org/10.1007/978-3-030-66891-4_11

[12] H. Vardhan, N. Surana, and B. Tripathy, *Named-Entity Recognition for Legal Documents*, 01 2021, pp. 469–479.

[13] T. Luiggi, L. Soulier, V. Guigue, S. Jendoubi, and A. Baelde, "Dynamic named entity recognition," 02 2023.