SemEval-2022 Task 6B
Legal Named Entities Extraction (L-NER)

Aditya Pandey, Shaily Goyal, Vedant Arvind Kumbhare
aditya.pandey@colorado.edu, shgo4037@colorado.edu, veku9185@colorado.edu

University of Colorado Boulder

Abstract

The development of legal artificial intelligence applications requires the identification of identified entities from legal texts as a fundamental building element. Compared to frequently used named entities like Person, Organization, and Location, named entities in legal texts differ somewhat and are more precise. The primary goal is to train a model for Named Entity recognition on Indian Court documents, specifically the preamble section.

1 Introduction

The ability to quickly and extensively use jurisprudence makes information retrieval and extraction critical challenges in modern legal practice. When creating their arguments for a case, lawyers are more likely to generate strong reasoning if they are given relevant instances to draw upon. It is also reasonable to assume that cases will be settled more definitively if a strong legal precedent is presented, even at an initial stage of the legal procedure. The fact that more and more technology solutions are being created in this field demonstrates its viability and importance. In this environment, open-source tools and resources are crucial to ensuring legal access is equitable.

The field of legal artificial intelligence (LegalAI) focuses on using artificial intelligence technologies, particularly natural language processing, to assist with work in the legal industry. This helps the judiciary process by relieving legal experts of a tangle of paperwork and has recently attracted increased interest from both AI academics and legal practitioners. While NLP scholars focus primarily on data-driven and embedding approaches, legal practitioners frequently consider ways to address problems using rule-based methodologies.

In this paper, we are training a Named Entity Recognition model. We create a pipeline that processes the text. The pipeline consists of 'tok2vec' and 'NER' components. The tok2vec component encodes the text into a vector representation and a transition-based parser that predicts the named entities. The model is trained on the training corpus and evaluated on the development corpus. The model weights are saved in the output directory which is then used for predictions. This model is evaluated based on the F1 score for named entities. The F1 score is the harmonic mean of the recall and precision of the named entities.

2 Background

The legal field has not given NER much attention up to this point, and the created techniques are uneven in terms of the methodology and datasets they employ. Regarding the choice of relevant semantic categories from the legal domain, there

is 4479 disagreement across the linked studies. Furthermore, it doesn't seem that there are any corpora or datasets of legal documents with annotated identified entities, which is certainly a barrier to the creation of data-driven NER classifiers.

Previous work on Legal-NER has focused on the development of specialized models and resources to improve the performance of NER systems in the legal domain. For example, researchers have developed legal corpora annotated with named entities, such as the Juris- Corpus (Bruns et al., 2011) and the Legal Entities Corpus (Zhou et al., 2015), which have been used to train and evaluate NER systems for legal text. Then we move on to the Named Entity Recognition models that are particular to the Indian judicial system. Legal NER: A Review of the State of the Art, by L. Gatti, S. Rizzo, and R. Navigli (2015), provides an overview of NER approaches applied to legal texts. Named Entity Recognition in the Legal Domain: A Review, by S. Kaur, A. Khosla, and A. K. Singh (2016), reviews different NER techniques applied to legal texts and discusses the challenges and limitations of existing approaches.

'Named Entity Recognition in Indian court judgments' by Prathamesh Kalamkar et al. provides a good dataset and baseline model for this project. Other studies have proposed the use of domain-specific techniques, such as the incorporation of legal ontologies (Xu et al., 2012) and the use of domain-specific word embeddings (Guo et al., 2016), to improve the performance of L-NER systems. Additionally, some researchers have explored the use of neural network models for L-NER, such as convolution neural networks (CNNs) (Liu et al., 2018) and recurrent neural networks (RNNs) (Xu et al., 2019), which have demonstrated promising results.

In specialized areas, where grammar may be more uniform, the range of topics covered may be less, words may have fewer senses, etc., the full potential of bigger, computationally more expensive models may not be essential. Therefore, 'A Named Entity Recognition Framework for Indian Legal Texts', by S. Bhatia and P. Aggarwal (2017), presents a NER framework for Indian legal texts and evaluates its performance on a small dataset. Whereas, A Study of Named Entity Recognition in Legal Texts, by M. J. Law, P. L. T. Loh, and C. C. Ong (2017), presents a study of NER in legal texts using a large dataset of Singaporean court judgments. A NER based model for Indian texts with large datasets whose evaluation performance is evaluated in the large dataset is shown in the paper Named Entity Recognition for Indian Legal Texts, by S. Kaur, A. Khosla, and A. K. Singh (2018).

'Named Entity Recognition as Dependency Parsing'(Yu et al., 2020) discusses the use of dependency parsing for NER. They propose to use dependency parsing for NER in complex situations that outperform the previous approaches. Additionally, 'LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention' (Ikuya Yamada et al., 2020) presents a new method for generating deep contextualized entity representations using entity-aware self-attention. They can also handle long-term dependencies and capture the context-dependent nature of entities.

3 System Overview

We have used the single-Tok2Vec-component shared setup. All other components, including the entity recognizer, use a Tok2VecListener layer as their model's tok2vec parameter, which connects to the tok2vec component model. [1]

[1] Project Repo: https://github.com/callingmedic911/legal_NER

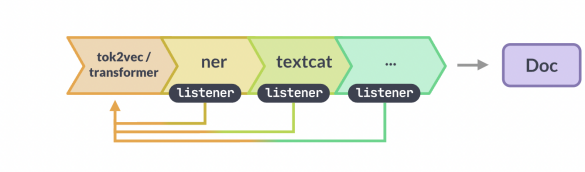

Figure 2: NER Architectures

3.1 Training Procedure

Using the spaCy library's configuration file, we described the natural language processing (NLP) pipelines. The pipeline includes two components: "tok2vec" and "ner". The "tok2vec" component uses a maxout window encoder to convert tokens into vectors, while the "ner" component uses a transition-based parser to identify named entities in text. The configuration specifies details such as the language of the text, the batch size, and the optimizer used for training. It also defines the corpora used for training and development, as well as the parameters for training and evaluation.

At the start of training, the Tok2Vec component will obtain a reference to the listener tiers in the pipeline. After analyzing a batch of documents, it will send its forecasts to the listeners so they may use them again when they are eventually called.

Here are the processes to prepare data and further use the output from the trained model to show results visually:

DATA-PREPARATION: The function *guess_preamble_end*() takes in a text string and a Spacy NLP model and uses heuristics and pattern matching to guess the end of the preamble in the text. The function uses the NLP model to tokenize and parse the text and then looks for sentences with verbs and specific patterns to determine the end of the preamble. If the preamble end cannot be determined using these methods, the function uses keywords to search for the end of the preamble. The character offset at the end of the preamble is then returned.

POST-PROCESSING: It is a Python script that processes a legal document using Natural Language Processing (NLP) techniques to identify entities such as people and organizations mentioned in the document. It then uses the nltk package to calculate the Levenshtein distance between pairs of names to group them together as likely referring to the same person or organization. The script also uses regular expressions to identify references to prior cases, and groups these references together based on the similarity of the names of the parties involved in the case.

LEGAL-NER: The spacy components needed for legal NER include the spaCy library, the displaCy library, the re library, the seperate_and_clean_preamble function, the get_text_from_indiankanoon_url function, the get_sentence_docs function, the postprocessing function, the get_csv function, the Span class, the time library, the get_unique_precedent_count function, the get_unique_statute_count function, and the get_unique_provision_count function. These components are used to extract entities from a given judgment text, process the preamble and judgment text, combine the preamble and judgment text, and perform postprocessing. The extracted entities can then be saved as a CSV file or displayed using displaCy.

4 Experimental setup

The model uses a pipeline [2] with the "tok2vec" [3] and "ner" components and uses a batch size of 1000. The tokenizer used is the "spacy.Tokenizer.v1" [4] tokenizer. The NER component uses a transition-based parser for the model, with a hidden width of 64 and maxout pieces of 2. The tok2vec component uses a maxout window encoder with a width of 256, depth of 8, and maxout pieces of 3. The training parameters specify a dropout rate of 0.1, accumulate gradient of

[2] https://spacy.io/models/en_core_web_sm
[3] https://spacy.io/api/tok2vec
[4] https://spacy.io/usage/v2-3

1, and a maximum number of steps of 20000. The evaluation frequency is set to 200, and the optimizer used is Adam with a learning rate of 0.001. The optimizer used is Adam. The model is evaluated based on the F1 score for the named entities.

5 Results & Discussion

E	#	$ENTS_F$	$ENTS_P$	$ENTS_R$	SCORE
0	0	0.00	0.00	0.00	0.00
0	200	10.56	18.68	7.36	0.11
0	400	16.19	18.38	14.46	0.16
...					
10	9200	78.92	79.90	77.97	0.79
11	9400	78.43	81.36	75.69	0.78
11	9600	78.32	77.33	79.33	0.78

The table above shows the training after tuning the parameter.

In our experiments, we trained a named entity recognition model on a corpus of Indian court documents. We evaluated the performance of the model on a held-out test set, and obtained a F1 score of 0.78.

The results of our model show that it is effective at identifying named entities in legal texts. This is a crucial building block for developing legal AI applications, as the ability to accurately identify and classify named entities is essential for tasks such as legal document analysis and case prediction.

There are several factors that may have contributed to the performance of our model. First, we used a transition-based parser, which is well-suited for the task of named entity recognition in legal texts. This parser, which uses a classifier to predict the named entities in the text, is able to capture the complex syntactic structure of legal documents and accurately identify named entities.

Second, we used a tok2vec component to encode the text into a vector representation. This allowed the model to capture the semantics of the text and improve its performance on the named entity recognition task.

Overall, the results of our experiments demonstrate the effectiveness of our named entity recognition model on Indian court documents. In future work, we plan to extend this model to other legal domains and continue to improve its performance. We also plan to explore other natural language processing techniques that may further improve the model's ability to identify named entities in legal texts.

6 Post-Processing of Named Entities

The following details of post-processing are inspired by research paper 'Named Entity Recognition in Indian court judgments' (Kalamkar et al., 2022)

For data preparation, we first perform preprocessing, coreference resolution, and post-processing. The preprocessing part of the code is responsible for cleaning the text and removal of unwanted characters and symbols from the text.

The coreference resolution is a part where coreferences are resolved. (The process of locating all phrases in a text that refer to the same thing is known as coreference resolution).

The problem with legal information extraction starts with confusion that is created when judges mention someone in short. For example often times judges while citing the names of the petitioner or lawyer only mention the first or the last name of the organization or the person. " Next judgment to be noticed is Constitution Bench judgment of Tikayat Shri Govindlalji Maharaj etc. Vs. State of Rajasthan and Others". But later the judge uses the reference to 'Tilkayat' as shortened form of the petitioner. "It was contended by Tilkayat that the idol of Shri Shrinathji in the Nathdwara Temple and all the properties pertaining to it were his private properties". Here the petitioner 'Tilkayat' is referenced by the legal NER model as OTHER-PERSON. But, Tikayat Shri Govindlalji Maharaj is the petitioner.

In such a case, the extracted cluster entities are compared to the names and citations. The *precedent_cluster* contains the previous or precedent entities. The keyword supra is checked. Then

the concerned entity is matched against the extracted precedent names and if a suitable match is found, the entity type is changed to Precedent. Thus, the name 'Tilkayat' is referred to as 'Precedent'. After grouping all the precedent entities, the longest entity is defined as the cluster head. In this example, 'Tikayat Shri Govindlalji Maharaj etc. Vs. State of Rajasthan and Others' is labelled as Cluster head.

The model can at times, tag the same entity as some other entity. For example "when Dr. Rajeev Dhavan, learned senior counsel appearing for the appellants", is a statement the judge writes, whereas later he mentions 'The above submission of Dr. Dhavan was opposed by learned counsel appearing for the respondents.' Here, the NER model depicts 'Dhavan' as ORG or $OTHER\_PERSON$ where he should be depicted as 'LAWYER'. This creates a problem where an important entity like the 'LAWYER' is defined instead as 'ORG'.

For this $other\_person\_coref\_res$ function is used. Here, the ORG is matched with all the $known\_person$ i.e. PETITIONER, JUDGE, RESPONDENT, WITNESS, and LAWYER. If a match is found, the entity is overwritten with the correct one. In this case, Dhavan is written as LAWYER.

Judges often write the statues in short form since it can be long and tedious to repetitively write same statue. For example the judge writes "The Mathadipati of Shirur Mutt filed a writ petition in Madras High Court challenging various provisions of Madras Hindu Religious and Charitable Endowments Act, 1951." but the judge later mentions "Challenge to the Act was on various grounds including the ground that provisions". Here the 'Hindu Religious and Charitable Endowments Act, 1951' is referred to as 'act'. For the recognition of statues, we search the statue cluster where it is matched against the extracted statues. If the match is found, the statue full form is added to the cluster.

An explicit reference to a provision and the relevant legislation can occur in the same phrase. For instance, "The Indian Penal Code Section 321 is Voluntarily causing hurt..." When just the provision is stated and the related legislation is inferred from the context, the mapping between the provision and statute might sometimes be implicit. This can be found by searching for STATUTE entities. These statute referents are inserted into the cluster of statutes, whose head is the full name of the legislation. The relevant complete form is included in the statutes collection if a match is discovered.

7 Conclusion & Future Directions

We have presented the model for named entity recognition on Indian court documents, specifically the preamble section. Our model, which consists of a tok2vec component and a transition-based parser, was trained on a corpus of legal texts and evaluated based on the F1 score for named entities. The results show that our model is effective at identifying named entities in legal texts, which is a crucial building block for developing legal artificial intelligence applications.

We plan to extend this model to other legal domains and continue to improve its performance. We also plan to explore other natural language processing techniques, such as contextualized word embeddings, that may further improve the model's ability to identify named entities in legal texts. Additionally, we plan to develop additional legal AI applications that can assist with tasks such as legal document analysis and case prediction.

The optimized scores after iteration are possible to be achieved if stronger computing resources are used to implement this project. With better resources, the F1 score may improve.

8 Acknowledgements

We would like to acknowledge the work of Prathamesh Kalamkar et al. on their paper "Named Entity Recognition in Indian court judgments", which served as a valuable resource for this task. We

would also like to thank our professor and the teaching staff at the University of Colorado Boulder for their guidance and support.

9 References

Embeddings, Transformers and Transfer Learning: spacy.io/usage

Oshin Agarwal, Yinfei Yang, Byron C Wallace, and Ani Nenkova. 2021. Entity-switched datasets: An approach to auditing the in-domain robustness of named entity recognition models. arXiv preprint arXiv:2004.04123.

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7747– 7763, Online. Association for Computational Linguistics.

Jason P.C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. Transactions of the Association for Computational Linguistics, 4:357–370.

Nigel Collier and Jin-Dong Kim. 2004. Introduction to the bio-entity recognition task at JNLPBA. In Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP), pages 73–78, Geneva, Switzerland. COLING.

Ronan Collobert, Jason Weston, Leon Bottou, Michael ´ Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. Journal of machine learning research, 12(ARTICLE):2493–2537.

Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Emerging crosslingual structure in pretrained language models. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 6022– 6034, Online. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. Transactions of the Association for Computational Linguistics, 8:64–77.

Prathamesh Kalamkar, Aman Tiwari, Astha Agarwal, Saurabh Karn, Smita Gupta, Vivek Raghavan, and Ashutosh Modi. 2022. Corpus for automatic structuring of legal documents. In Proceedings of the Language Resources and Evaluation Conference, pages 4420–4429, Marseille, France. European Language Resources Association.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,pages 260–270.

Bruns, A., Burgess, J. (2016). Methodological Innovation in Precarious Spaces: The Case of Twitter. In: Snee, H., Hine, C., Morey, Y., Roberts, S., Watson, H. (eds) Digital Methods for Social Science. Palgrave Macmillan, London. https://doi.org/10.1057/9781137453662$_2$

Zhou, D., Chen, L., He, Y. (2015). An Unsupervised Framework of Exploring Events on Twitter: Filtering, Extraction and Categorization. Proceedings of the AAAI Conference on Artificial Intelligence, 29(1). https://doi.org/10.1609/aaai.v29i1.9526

Legal NER: A Review of the State of the Art, by L. Gatti, S. Rizzo, and R. Navigli (2015)
Named Entity Recognition in the Legal Domain: A Review, by S. Kaur, A. Khosla, and A. K. Singh (2016)
Xu, Z., Ni, Y., He, W. et al. Automatic extraction of OWL ontologies from UML class diagrams: a semantics-preserving approach. World Wide Web

15, 517–545 (2012). https://doi.org/10.1007/s11280-011-0147-z
Yantuan Xian, Fa Shao, Jianyi Guo, Lanjiang Zhou, Zhengtao Yu, and Wei Chen. 2016. Using deep belief networks to extract Chinese entity attribute relation in domain-specific. Int. J. Comput. Sci. Math. 7, 2 (May 2016), 144–155. https://doi.org/10.1504/IJCSM.2016.076422

Yamashita, R., Nishio, M., Do, R.K.G. et al. Convolutional neural networks: an overview and application in radiology. Insights Imaging 9, 611–629 (2018). https://doi.org/10.1007/s13244-018-0639-9

[Named Entity Recognition as Dependency Parsing](https://aclanthology.org/2020.acl-main.577) (Yu et al., ACL 2020)

[LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention](https://aclanthology.org/2020.emnlp-main.523) (Yamada et al., EMNLP 2020)

Benikova, D., Yimam, S.M., Santhanam, P., Biemann, C. (2015). GermaNER: Free Open German Named Entity Recognition Tool. German Society for Computational Linguistics.

Cristian Cardellino, Milagro Teruel, Laura Alonso Alemany, Serena Villata. A Low-cost, High-coverage Legal Named Entity Recognizer, Classifier and Linker . ICAIL-2017 - 16th International Conference on Artificial Intelligence and Law, Jun 2017, Londres, United Kingdom. pp.22. ⟨hal-01541446⟩

[LEGAL-BERT: The Muppets straight out of Law School](https://aclanthology.org/2020.findings-emnlp.261) (Chalkidis et al., Findings 2020)
Zhong, H., Xiao, C., Tu, C., Zhang, T., Liu, Z., Sun, M. (2020). How Does NLP Benefit Legal System: A Summary of Legal Artificial Intelligence. Annual Meeting of the Association for Computational Linguistics.

Leitner, Elena Rehm, Georg Schneider, Julián. (2020). A Dataset of German Legal Documents for Named Entity Recognition.

Luz de Araujo, Pedro Henrique de Campos, Teofilo Oliveira, Renato Stauffer, Matheus Couto, Samuel De Souza Bermejo, Paulo. (2018). LeNER-Br: A Dataset for Named Entity Recognition in Brazilian Legal Text: 13th International Conference, PROPOR 2018, Canela, Brazil, September 24–26, 2018, Proceedings. 10.1007/978-3-319-99722-$3_3$2.

Henrique, Pedro Luz de Araujo, Pedro Henrique de Campos, Teofilo Oliveira, Renato Stauffer, Matheus Couto, Samuel De Souza Bermejo, Paulo. (2018). LeNER-Br: A Dataset for Named Entity Recognition in Brazilian Legal Text.

Paul, S., Goyal, P., Ghosh, S. (2022). LeSICiN: A Heterogeneous Graph-Based Approach for Automatic Legal Statute Identification from Indian Legal Documents. Proceedings of the AAAI Conference on Artificial Intelligence, 36(10), 11139-11146.

Dozier, C.C., Kondadadi, R., Light, M., Vachher, A., Veeramachaneni, S., Wudali, R. (2010). Named Entity Recognition and Resolution in Legal Text. Semantic Processing of Legal Texts.

Kalamkar, Prathamesh Agarwal, Astha Tiwari, Aman Gupta, Smita Karn, Saurabh Raghavan, Vivek. (2022). Named Entity Recognition in Indian court judgments. 10.48550/arXiv.2211.03442.