Ginn-Khamov at SemEval-2023 Task 6, Subtask B: Legal Named Entities Extraction for Heterogenous Documents

Michael Ginn and **Roman Khamov**
University of Colorado
michael.ginn@colorado.edu, roman.khamov@colorado.edu

Abstract

This paper describes our submission to SemEval-2023 Task 6, Subtask B, a shared task on performing Named Entity Recognition in legal documents for specific legal entity types. Documents are divided into the preamble and judgement texts, and certain entity types should only be tagged in one of the two text sections. To address this challenge, our team proposes a token classification model that is augmented with information about the document type, which achieves greater performance than the non-augmented system.

1 Introduction

The goal of the Legal NER subtask of the LegalEval task is to perform named entity recognition (NER) on English legal texts (Modi et al., 2023). We are interested in detecting entities specific to the legal domain, such as case numbers, as well as differentiating between legal roles for people, such as lawyers and judges. This information can then be used for downstream tasks such as information retrieval.

While industry-standard NER systems such as spaCy's transition-based parser (Honnibal and Johnson, 2015) achieve state-of-the art performance on most standard NER tasks, it can be desirable to have a customizable system for a domain task (Tang et al., 2017). Custom NER systems can support domain-specific entity types and learn to detect entities in structured domain-specific documents.

Our primary system utilizes a pretrained language model with a token classification head.

The data provided by Kalamkar et al. (2022) consists of annotated sentences from Indian English court judgement documents, which are divided into two sections, the preamble text and judgement text. We observed that while entities of certain classes appear in both the preamble and judgement texts, they should only be identified in one of the two document classes. To address this imbalance, we introduced augmentations that aided our system in making accurate predictions for each document class.

We discover that our system is able to achieve similar performance to the baseline on the validation data, and that certain augmentations allow the system to learn more quickly and accurately. Our system shows improvements for certain entity types, although certain entity types remain difficult to label.

Our code is available at https://github.com/michaelpginn/SemEval2023-LegalNER.

2 Background

2.1 Datasets

The dataset provided by (Kalamkar et al., 2022) consists of 11.0k training sentences and 1k validation sentences, labeled with fourteen different entity types. For example, in the following sentence, "Sri Raja Amareshwar Naik" should be tagged as a RESPONDENT.

> *No one was examined as witness on behalf of other respondents, including respondents 2 and* ***Sri Raja Amareshwar Naik.***

2.2 Evaluation Metrics

The systems are primarily evaluated using the average F1 score over all entity labels, where entities must match exactly to be considered correct. Additionally, we compute the F1, precision, and recall scores for each entity to assess model weaknesses.

2.3 Baseline System

The best-performing baseline system (Kalamkar et al., 2022) uses a spaCy transition-based parser (Honnibal and Johnson, 2015) with the pretrained RoBERTa model (Liu et al., 2019) used to provide contextualized embeddings. Kalamkar et al. (2022)

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 420–425

also experimented with different pretrained language models such as LegalBERT (Chalkidis et al., 2020) and InLegalBERT (Paul et al., 2022), as well as a Transformer finetuning approach built with the T-NER library (Ushio and Camacho-Collados, 2021). Critically, the baseline system does not finetune the underlying transformer, but only trains the transition-based parser.

3 System Overview

Our base NER system uses a pretrained transformer encoder with a token classification head on top, fine-tuned on the training data (Figure 1). Finetuning pretrained transformers has proven very effective for NER (Lothritz et al., 2020). We experimented with both RoBERTa (Liu et al., 2019) and LegalBERT, a BERT model trained from scratch on legal texts (Chalkidis et al., 2020), for the encoder. We use IOB tagging, where each entity type has an *inside*, *outside*, and *beginning* tag, resulting in 29 total output tags.

3.1 Data Augmentation for Heterogeneous Sentences

One key challenge our base system ran into is that entities are tagged differently depending on whether they came from the preamble or judgement text. For example, case numbers appear in both types of document, but are only tagged in the judgement text, causing our models to have high recall but low precision for these entities, as there were many false positives. Thus, our system must learn to recognize entities in a heterogeneous manner, considering the source document type of the sentence.

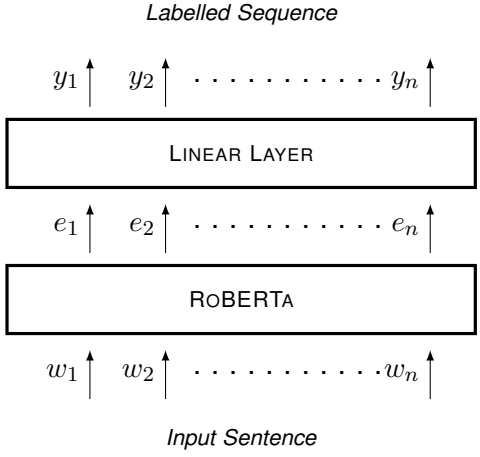

Figure 1: Fine-Tuned Model Architecture

While the provided data divides sentences into the preamble and judgement text, there is no guarantee that our model will have access to this. Furthermore, a transformer should hypothetically be able to learn this distinction, but it is likely that this is too complex to infer with the amount of provided data.

Thus, we trained a second model to predict the source of a sentence based on its raw text. This model used LegalBERT (Chalkidis et al., 2020) as an encoder with a one-node fully-connected layer on top for binary classification. The classification model's performance is presented in Table 1.

	F1	P	R
Classification Model	99.1	98.8	99.4

Table 1: Sentence classification model performance

This trained model is used to augment the NER model in three distinct approaches.

- In the first approach (Figure 2), we use the classification model to predict the class for each sentence, and add one of two special tokens to the end of the sequence indicating the document type. We resize the pretrained encoder for the new tokens and run training and inference as usual. Because the class information is explicitly present in the input sequence, the transformer should learn when to classify certain entity types more easily.
- In the second approach (Figure 4), we design a twin-transformer model with two identical NER models. For each batch of sentences, we use the predictions from the classification model to split the batch into preamble and judgement sentences, and run each partial batch through one of the two models, aggregating the loss. Because each model only sees sentences of one type, it should learn which entities to label.
- In the third approach (Figure 3, we modify the token classification linear layer so it also accepts a parameter indicating the class of the sentence, provided by the classifier predictions. Again, this context should help the model learn the differences between documents, but unlike the first model, the information is available at the last step rather than the first.

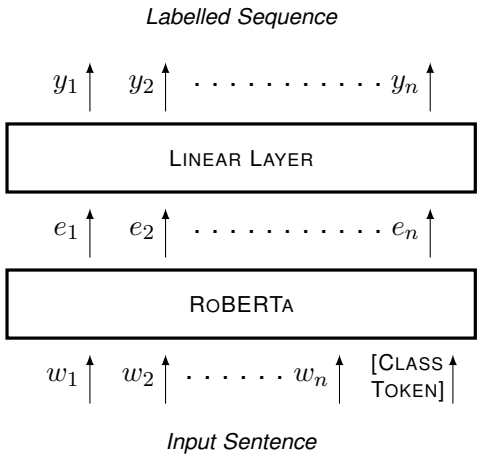

Figure 2: Append Sentence Classification Token

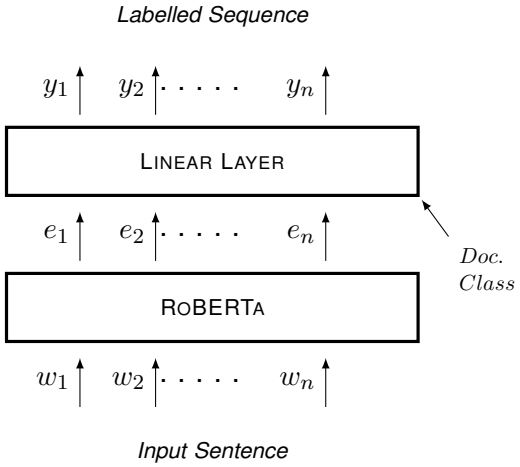

Figure 3: Augmented Linear Layer

4 Experimental Setup

4.1 Preprocessing

In order to do IOB tagging, we first load the data from the spaCy binary format, which is tokenized into words. For each sequence of tokens that are tagged as an entity, we set the first label to the B-CLASS tag, such as B-LAWYER, and set all others to the I-CLASS tag, such as I-LAYWER. All other tokens are labelled with O.

Next, we tokenize and encoder using the appropriate byte-pair encoder for our pretrained LM. We map each subword token to the appropriate IOB tags and ignore all special tokens.

A few of the validation rows did not have entities tagged correctly, so these were omitted.

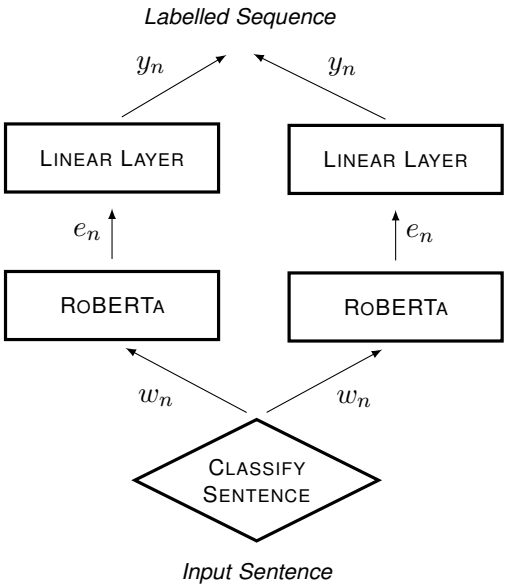

Figure 4: Twin Model

4.2 Experiments

First, we trained and evaluated the baseline system and a few small variations, for four different experiments. Models were trained with the training parameters specified in Kalamkar et al. (2022). In addition to the baseline, we trained with LegalBERT instead of RoBERTa, with dropout increased from 0.1 to 0.3, and with the hidden state size increased from 64 to 128.

Next, we trained five custom models using fine-tuned transformers. First, we trained two models with LegalBERT and RoBERTa, and selected the higher performing model for augmentation; in our experiment this was RoBERTa. Next, we trained and evaluated models with the three augmentations mentioned: a model where the sentence class was added as an input token, a twin model where sentence class was used to pick one of the two sub-models, and a model where the sentence class was provided to the final linear layer.

Each model was trained on all of the training data and evaluated using all of the remaining validation data. Models were trained for 40 epochs with a batch size of 64 and 3 gradient accumulation steps. Training used the AdamW (Loshchilov and Hutter, 2017) optimizer with learning rate of 2E-5, weight decay of 0.01, beta1 of 0.9, beta2 of 0.999, and epsilon of 1E-8. Models were trained on an Nvidia RTX A6000; training for each model took around 3 hours.

Model		F1	P	R
spaCy Baseline	Baseline (RoBERTa)	90.3	90.3	90.4
	Baseline (LegalBERT)	89.3	89.2	89.4
	Baseline (RoBERTa) + higher dropout (0.3)	90.1	90.1	90.1
	Baseline (RoBERTa) + larger hidden state (128)	89.0	89.1	88.9
Fine-Tuned Transformer	LegalBERT base	85.6	83.1	88.1
	RoBERTa base	87.3	86.0	88.7
	RoBERTa + sentence class. token	**89.9**	**88.6**	91.2
	RobERTa + twin model	89.8	88.4	**91.3**
	RoBERTa + aug. linear layer	89.6	88.5	90.7

Table 2: Average performance of all models on validation set

5 Results

The overall performance for each model on the validation data is listed in Table 2. All of the systems had high performance scores, indicating this task is very tractable.

Our best system achieved an F1 of 72.7 on the test data, ranking 12 of 17. As this performance is significantly worse than our validation set, this indicates that either our system is overfit to the validation data or the validation and test data have significantly different distributions.

The former seems unlikely, since we only tested the models listed against the validation data, and all of them performed well on both the training data and unseen validation data. Thus, we believe that the test data must have been different than the training and validation data in some way, such as having different distribution of entity types (maybe the patterns we observed with entities in preamble and judgement texts weren't consistent?). We do observe that the test data has a much greater proportion of sentences that were truncated by our model input length, which could be partly responsible for the error.

5.1 Effect of Fine-tuning

None of the variations to the baseline spaCy model improved performance. The fine-tuned models did not quite outperform the best spaCy baseline model, although the difference was very small. Generally, the fine-tuned models tended to have slightly worse precision and slightly better recall. This indicates that while a simple fine-tuned transformer model doesn't perform significantly better than the SOTA solution, it is very competitive. One possible solution to beat the SOTA would be to constrain the output of the fine-tuned model using a conditional random field (CRF) layer on top, while continuing to fine-tune the entire model.

We hypothesize that the SOTA would have a larger performance disparity on a general NER task, where fine-tuning a transformer would not provide much benefit.

The fine-tuned systems tended to have higher recall than precision. One reason for this behavior is that, as mentioned previously, certain entity types appear but should not be labelled in either the preamble or judgement, leading to many false positives.

5.2 Effect of LegalBERT

While we initially hypothesized that using a domain specific model such as LegalBERT would improve performance over a general-purpose model, the LegalBERT models underperformed for both the baseline model and our fine-tuned model. One possible explanation is that the benefit of pretraining on legal data did not outweigh the advantage of the RoBERTa architecture over the standard BERT architecture. Another possibility is that the data used to pretrain LegalBERT, which consisted of European legal documents, was not similar enough the Indian legal documents used in this task.

	spaCy Baseline	RoBERTa Base	+ Class. Token
LAWYER	96.4	97.3	**98.1**
DATE	**98.4**	96.6	**98.4**
ORG	**73.9**	62.5	70.0
GPE	81.5	73.8	**83.1**
STATUTE	**93.9**	91.1	92.2
PROVISION	93.8	92.4	**95.8**
PRECEDENT	77.8	73.0	**79.6**
CASE NUMBER	**80.3**	76.6	76.0
WITNESS	**97.4**	91.5	95.8
OTHER PERSON	**93.4**	88.2	91.3

Table 3: Entity-level F1 on validation data

5.3 Effect of Augmentation Techniques

All of the augmented models outperformed the base model by 2+ points. As predicted, the base model had difficulty providing labels differently based on document type.

Table 3 indicates the performance for entities that only appear in one of the two document types. The best-performing augmented model outperforms the baseline on all but one entity type, confirming that augmentation provided benefits. However, the augmented model only outperformed the spaCy baseline on some entities.

Of the augmented models, the model that appended a classification token performed best and the model that augmented the linear layer performed the worst. The best model had an F1 of 89.9, which is only 0.4 less than the SOTA.

The closeness between all augmented models indicates that as long as the document class information is explicitly available, it does not matter when it is provided to the model.

The twin model achieved the highest recall of any model including the SOTA.

5.4 Error Analysis

Our best model has particular difficulty with labeling ORG entities, and also struggles at STATUTE, CASE NUMBER, WITNESS, and OTHER PERSON labels. One likely reason for this is that these entities are either similar to the surrounding text, or similar to other entity types.

In order to better distinguish entities from surrounding text, we can augment the data with features such as word shape and other character-level information, which spaCy uses. Song et al. (2021) finds that for biomedical NER, extracting character-level features is an effective strategy.

In order to distinguish between similar entity types, we could integrate the use of a knowledge base for relevant entities, as in Tedeschi et al. (2021).

6 Conclusion

In this paper, we describe a fine-tuned transformer model for named entity recognition in the legal domain, which fails to outperform the SOTA, although producing very similar results. Using a custom transformer architecture is desirable in domain-specific tasks, as it allows customization to the task at hand.

We demonstrate that in situations where documents should not all be labelled in the same manner, augmenting data using a trained classification model can improve performance. To this end, the simplest and most effective solution is to append a special token to the sequence indicating document class. We also demonstrate that domain-specific pretrained language models do not necessarily offer performance benefits over general-purpose models. Future work could involve augmenting input data further using information such as word shape.

References

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. LEGAL-BERT: The Muppets straight out of Law School. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online. Association for Computational Linguistics.

Matthew Honnibal and Mark Johnson. 2015. An Improved Non-monotonic Transition System for Dependency Parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1373–1378, Lisbon, Portugal. Association for Computational Linguistics.

Prathamesh Kalamkar, Astha Agarwal, Aman Tiwari, Smita Gupta, Saurabh Karn, and Vivek Raghavan. 2022. Named Entity Recognition in Indian court judgments. ArXiv:2211.03442 [cs].

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization.

Cedric Lothritz, Kevin Allix, Lisa Veiber, Jacques Klein, and Tegawendé François D Assise Bissyande. 2020. Evaluating pretrained transformer-based models on the task of fine-grained named entity recognition. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3750–3760.

Ashutosh Modi, Prathamesh Kalamkar, Saurabh Karn, Aman Tiwari, Abhinav Joshi, Sai Kiran Tanikella, Shouvik Guha, Sachin Malhan, and Vivek Raghavan. 2023. SemEval-2023 Task 6: LegalEval: Understanding Legal Texts. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, Toronto, Canada. Association for Computational Linguistics (ACL).

Shounak Paul, Arpan Mandal, Pawan Goyal, and Saptarshi Ghosh. 2022. Pre-training transformers on indian legal text.

Bosheng Song, Fen Li, Yuansheng Liu, and Xiangxiang Zeng. 2021. Deep learning methods for biomedical named entity recognition: a survey and qualitative comparison. *Briefings in Bioinformatics*, 22(6):bbab282.

Siliang Tang, Ning Zhang, Jinjiang Zhang, Fei Wu, and Yueting Zhuang. 2017. NITE: A neural inductive teaching framework for domain specific NER. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2652–2657, Copenhagen, Denmark. Association for Computational Linguistics.

Simone Tedeschi, Valentino Maiorca, Niccolò Campolungo, Francesco Cecconi, and Roberto Navigli. 2021. WikiNEuRal: Combined neural and knowledge-based silver data creation for multilingual NER. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2521–2533, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Asahi Ushio and Jose Camacho-Collados. 2021. T-NER: An all-round python library for transformer-based named entity recognition. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 53–62, Online. Association for Computational Linguistics.