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

Anuradha Mysore Ravishankar, Spriha Awasthi, Nitish Venkatesh Septankulam Ramakrishnan

University of Colorado Boulder

Abstract

SemEval-2022 Task 6 Part B, presents a shared task that extracts the Legal Named Entities from text. Legal documents text contains entities which are much different from plain text documents, such as names of petitioner, respondent, court, statute, provision, precedents, etc. Because these entity types are not understood well by the standard pretrained named entity recognizers (NERs), there is a need to develop a Legal NER system. Here we experiment and attempt to build a Legal NER by fine tuning pretrained BERT (base-cased), DistilBERT and XLNet. All models give 95-96% weighted accuracy scores but with limitations due to class imbalance and some entities remain hard to classify. This report summarizes the results and findings of our shared task.

Keywords: Named Entity Recognition, Legal Named Entity Recognition, Legal NER, Bidirectional Encoder Representations from Transformers, DistilBERT, XLNet, Natural Language Processing

Introduction

There are over 47 million cases currently pending in courts across various judiciary systems in India. Of them, 87.4% are pending in subordinate courts, and 12.4% in High Courts. Various cases are yet to be addressed and pending for over a decade, the potential for technical solutions to streamline the process is significant. With the SemEval-2022 Task 6 Part B, we address this issue by attempting to build a model for Named Entity Recognition (NER) that is specifically trained for tagging legal texts and entities. The conventional interpretation for this project is that using legal-NER allows one to parse texts from documents, and increase the speed and scale of content collection by extracting data such as case numbers, petitioner, respondent, etc., and expedite cases which in turn fast-tracks court hearings and provide value by streamlining the process.

Named Entity Recognition (NER) is one of the most popular data preprocessing techniques, where information extraction techniques are used to classify texts into “Named Entities” from unstructured texts. Words can be classified to predefined categories such as Person, Location, Organization, etc. Thus, the SemEval Task 6 Part B requires a text-to-text sequence generation, that is a set of words corresponding to a set of entities. To do this the basic approach we will take is to utilize the Bidirectional Encoder Representations from Transformers (BERT) which is a deep learning model in which every output element is connected to every input element, and the weightings between them are dynamically calculated based upon their connection. BERT’s key technical innovation is applying the bidirectional training of Transformer, a popular attention model, to language modeling. This is in contrast to previous

efforts which looked at a text sequence either from left to right or combined left-to-right and right-to-left training. Here we use the Base BERT model (cased), DistilBERT and XLNet models for tag generation. While the BERT base model is a good candidate to start with, we also chose DistilBERT as it is a small, inexpensive, and an efficient transformer model based on the BERT architecture. It uses knowledge distillation to reduce the size of the pre-training phase by 40%. Distillation is a technique used to compress large models and knowledge distillation allows us to use compression techniques where the previously mentioned compressed model is reproduced to behave like a large model. In addition XLNet has shown good improved performance over BERT in a range of NLP Based classification tasks and is used in our study.

The models perform with limitations due to class imbalance in the dataset and limitations from populating newer entries. The overall weighted accuracy recorded is 95%-96% for all models and macro-F1 scores between 0.28-0.35. The rest of the paper is organized as follows: Section 2 covers Background work, which covers research papers on entity recognition on legal texts. Section 3 covers "System Setup"; a consolidation of various steps and components involved in the work, Section 4 contains the "Experimental Setup" which details the data splits and hyperparameters used; followed by Section 6, "Results" detailing key findings and analysis; and finally we conclude with "Conclusion". Our source code is available at [11].

Background

Legal texts often have domain-specific entities. Therefore, the usage of a NER specific model is imperative. (Prathamesh et. Al, [2]) created a Named Entity Recognition in Indian Court Judgements with the use of "en_legal_ner_trf". The data obtained was split into train, dev, and test sets. The models are evaluated based on recall, precision, and F1 scores. For the SemEval-2022 task, we have similarly used the aforementioned scores to judge our model.

A myriad number of studies for NER approaches have been trained and evaluated on English texts from conventional domains. However, legal texts have long and complex sentences and often legal jargon is used for this domain specific vocabulary. Therefore, this makes them less efficient. ([9]). Furthermore, the dataset used for the project was incomplete. In our case, the test set was missing and old parsers were broken in logic after URLs updated their format. To combat this problem, we are employing our dev set as a test set and rather splitting the training set into train/validation. Additionally, the formalization of the NER task as a sequence generation is mentioned. Given an input sentence i.e. a sequence of tokens, the evaluation of that particular sequence is performed where every entity in the sequence is assigned a label from a predefined list of label types. Literature review also indicated best accuracies achieved of 49% in similar setups ([9]).

Christopher et al.[4] used statistical models with representative data that is processed at runtime. They used manually annotated data and analyzed the inherent ambiguity present. They used a balanced dataset with a few thousand samples in each category. Furthermore, they focused on captioned documents for their model. Iosif et al., [5] split the available data roughly in the ratio of

63.7% for training, 17.8% for validation, and 18.5% for testing. Out of Vocabulary words (OOV) were mapped into one single embedding: Unknown 'UNK.' Similar to SemEval-2022 and our report, they used precision, recall, and F1 scores. A macro-averaged F1 of 0.81 was obtained.

System Overview

The overall task can be modeled as a text-to-text generation problem which is solved well by encoder-decoder based architectures such as BERT and its variants. The main components of our system are:

1. Loading dataset and entity annotations.
2. Preprocessing and data cleaning.
3. Training models on the three different architectures as base models: BERT (base-cased), DistilBERT and XLNet.
4. Compare and evaluate the performance of trained models.

3.1 Dataset:

We have included the information on the dataset we used for our project below including the named entities of interest that are available through training dataset.

Table 1: Named Entities of interest

Named Entity	Extracted From	Description
COURT	Preamble, Judgment	Name of the court which has delivered the
PETITIONER	Preamble, Judgment	Name of the petitioners / appellants /
RESPONDENT	Preamble, Judgment	Name of the respondents/ defendants/
JUDGE	Premable, Judgment	Name of the judges from current case if extracted from preamble. Name of the judges of
LAWYER	Preamble	Name of the lawyers from
DATE	Judgment	Dates mentioned in the judgment
ORG	Judgment	Name of organizations mentioned in text
GPE	Judgment	Geopolitical locations.
STATUTE	Judgment	Name of the act or law mentioned
PROVISION	Judgment	Sections, sub-sections, articles, orders, rules
PRECEDENT	Judgment	All the past court cases referred in the judgement as precedent. Precedent consists of party
CASE_NUMBER	Judgment	All the other case numbers mentioned in the judgment (apart from precedent)
WITNESS	Judgment	Name of witnesses in
OTHER_PERSON	Judgment	Name of the all the person that are not included in petitioner,

Table 2: Counts of tags in different sections

Entity	Judgment Count	Preamble Count
COURT	1293	1074
PETITIONER	464	2604
RESPONDENT	324	3538
JUDGE	567	1758
LAWYER	-	3505
DATE	1885	-
ORG	1441	-
GPE	1398	-
STATUTE	1804	-
PROVISION	2384	-
PRECEDENT	1351	-
CASE_NUMBER	1040	-
WITNESS	881	-
OTHER_PERSON	2653	-
Total	**17485**	**12479**

The dataset used is divided in two parts, “Preamble” and “Judgment”. Table 2 represents a snippet of the Preamble which contains the names of parties, court, lawyers etc. The judgment text starts after the Preamble.

3.2 Preprocessing and cleaning challenges:

A big challenge for our work was posed by the data cleanliness and broken web crawlers that accompanied the datasets. This severely restricted our capacity in the time available to generate and produce more data. As a work around we attempted to do the following data cleaning and pre-processing:

- Parse the JSON format data available instead of raw data to create three lists - sentences, named entities and their labels.
- After parsing the JSON objects, we tokenize sentence list, and remove tokens that were irrelevant such as HTML Tags in between text. Spacy was used for primary library for this purpose.
- Assign non named entities with ‘other’ category for training purposes.
- Due to no access to test dataset, we use dev dataset for testing purposes and split train into train and validation set (80:20).

For this paper, we have train and dev dataset as the test dataset at the time of writing has not been released. We finally use it to compute metrics including accuracy, F1 scores and confusion matrix for all 13 classes of interest for three models. Results are discussed in next sections.

Some other challenges that were tackled are:

- Unstructured data modeling and formatting requirements made parsing harder.
- Label assignment at word level considering span start end for each named entity.
- Metric evaluation, length mismatches in prediction and actuals, additional preprocessing required to keep length consistent.
- Environment issues and hardware requirements, including GPU constraints.
- Depending on the court, the case number can be of varying length and format and may include letters, numbers, a combination of letters and numbers or even special characters which need cleaning up.

Experimental Setup

As described above, we use the dev set as a test set for our project and split the existing train set into a train/validation dataset in 80:20 ratio. This is because the test set is not yet available for analysis. This makes the train set size of 7,548, dev set size of 1,887 and test set size of 3,515. For each of the 3 models mentioned above, we train using the following hyperparameters:

1. Epochs: 20
2. Learning rate: 10^{-4}
3. Batch size: 32
4. Optimizer: Adam’s optimizer

Next we discuss our results in detail for the 3 models and shortcomings observed.

Results and Discussion

6.1 Approach:

Considering the training data as train data and dev data as test data; first, we imported the train data into a list. This is followed by separating the labels, NERs, and sentences. We proceed to find the start and end sequences. Custom methods were created for this task. Moving forward, we tokenize sentences while assigning labels. In addition to existing entities, we create a label “OTHER” to segregate texts that belong to zero existing labels. We proceed to create dataframes for the above and split the data into training and testing as mentioned above. We again create custom methods for each model and another method to accept model and data.

6.2 Evaluation Metrics:
Evaluation metrics evaluate the model's performance on train and test data. The SemEval-2022 task 6 Part B has mentioned the usage of F1-score. Our report contains the combination of F1-score, Precision, Recall, and Accuracy.

Our experimental results are tabulated below. The findings of the models' evaluation are reported in this section. Appropriate metric scores were employed along with their hyper parameter tuning.

Table 3: Tabulated results (in percentages %, all weighted average)

	BERT	XLNet	DistilBERT
Accuracy	96	95	95
Precision	96	96	96
Recall	96	95	95
F1-score	95	95	95

Table 4: Tabulated results (in percentages %, all macro avg)

	BERT	XLNet	DistilBERT
Precision	43	36	40
Recall	35	31	31
F1-score	35	28	31

6.4 Comparative Analysis and challenges:
The study here indicates the overall challenge of dealing with class imbalance problems. However, due to broken parsers in the dataset provided we were constrained to using only data available at hand as we previously mentioned in the System Design section. The difference in performance with respect to accuracy is negligible across all models but in terms of F1 scores, the BERT (base-uncased) performed better than the others. In addition to this, running baseline model in consumer machines was found to be demanding and therefore, F1-scores ranged from 0-0.82. This is misaligned and indicates a need for better data and is part of our future work goals.

Conclusion

Here we attempt to build from ground-up a legal named entity recognition system by extending state of the art models based on BERT (base-cased), DistilBERT and XLNet. After dealing with dataset challenges and preprocessing, BERT (base-cased) model seems to be best in terms of macro F1 score and overall all models struggle with limitations of class imbalance. An impressive weighted accuracy of 95% is accompanied by a macro F1 score of 0.35. This, however, falls only marginally behind with existing literature review results on similar datasets. As next steps, we would propose to deal with

the class imbalance by either fixing the crawler to generate more data or try more resampling techniques to improve performance. We hypothesize that lack of test data not being provided could be a key factor in not obtaining satisfactory results. Furthermore, multiple articles related to NER tagging in the legal domain were superficial. Also, scores of papers reported individual F1-measures while very few reported their overall scores. This is a disadvantage for us as our report primarily focuses on overall scores.

References

1. https://github.com/Legal-NLP-EkStep/legal_NER
2. Prathamesh Kalamkar, Astha Agarwal, Aman Tiwari et Al. "Named Entity Recognition in Indian court judgments"
3. Elena Leitner, Georg Rehm and Julian Moreno-Schneider, "Fine-Grained Named Entity Recognition in Legal Documents , " 2019.
4. Christopher Dozier, Ravikumar Kondadadi, Marc Light, Arun Vachher, Sriharsha Veeramachaneni, and Ramdev Wudali, "Named Entity Recognition and Resolution in Legal Text," 2010.
5. Iosif ANGELIDIS, Ilias CHALKIDIS, and Manolis KOUBARAKIS, "Named Entity Recognition, Linking and Generation for Greek Legislation," 2018
6. https://towardsdatascience.com/named-entity-recognition-with-bert-in-pytorch-a454405e0b6a
7. https://blog.finology.in/Legal-news/pending-cases-in-india
8. https://analyticsindiamag.com/guide-to-xlnet-for-language-understanding/
9. Stavroula Skylaki; Ali Oskooei; Omar Bari; Nadja Herger; Zac Kriegman, Legal Entity Extraction using a Pointer Generator Network, 2021 International Conference on Data Mining Workshops (ICDMW)
10. Juliana P. C. Pirovani, Elias Oliveira, "Studying the adaptation of Portuguese NER for different textual genres," 2021.
11. Source code: : https://github.com/Anu0705/NLP-Shared-Task