UO-LouTAL at SemEval-2023 Task 6: Lightweight Systems for Legal Processing

Sébastien Bosch, Louis Estève, Joanne Loo, Anne-Lyse Minard

UO-LouTAL at SemEval-2023 Task 6: Lightweight Systems for Legal Processing

Bosch Sébastien
University of Orléans (France)

Estève Louis
University of Orléans (France)

Loo Joanne
University of Orléans (France)

Minard Anne-Lyse
LLL-CNRS, University of Orléans (France)
anne-lyse.minard@univ-orleans.fr

Abstract

This paper presents the work produced by students of the University of Orléans Masters in Natural Language Processing program by way of participating in SemEval Task 6, LegalEval, which aims to enhance the capabilities of legal professionals through automated systems. Two out of the three sub-tasks available – Rhetorical Role prediction (RR) and Legal Named Entity Recognition (L-NER) – were tackled, with the express intent of developing lightweight and interpretable systems. For the L-NER sub-task, a CRF model was trained, augmented with post-processing rules for some named entity types. A macro $F_{\beta=1}$ score of 0.74 was obtained on the DEV set, and 0.64 on the evaluation set. As for the RR sub-task, two sentence classification systems were built: one based on the Bag-of-Words technique with L-NER system output integrated, the other using a sentence-transformer approach. Rule-based post-processing then converted the results of the sentence classification systems into RR predictions. The better-performing Bag-of-Words system obtained a macro $F_{\beta=1}$ score of 0.49 on the DEV set and 0.57 on the evaluation set.

1 Introduction

The ever-increasing volume of legal documents in populous countries such as India, for example, shows that automated systems may prove to be highly beneficial. Automating such processes reduces the need to perform unrewarding tasks such as manual information retrieval, thereby facilitating the work of legal professionals, while reserving more complex aspects of law and decisions for their intervention ; and shortens the overall time required for legal processing (Kalamkar et al., 2022b).

Two out of the three LegalEval 2023 (Modi et al., 2023) sub-tasks available were tackled: the Rhetorical Roles (RR) textual classification task (sub-task A) and the Legal-Named Entities Extraction (L-NER) task (sub-task B).

Systems for the sub-tasks were not developed in silo: some RR systems made use of data that was first pre-processed with L-NER system-generated annotations, which in some cases improved performance on the DEV set by approximately 2 points [1].

Some challenges encountered include (but are not limited to) handling complex and precise semantics developed over long spans of text in the RR sub-task; and understanding the subtleties distinguishing the five named entity types pertaining to persons (`JUDGE`, `OTHER_PERSON`, `PETITIONER`, `RESPONDENT`, `WITNESS`) for the L-NER sub-task.

Special attention was placed on developing lightweight and interpretable systems. The express focus on being lightweight was mainly to curb the carbon footprint of machine learning systems and improve widespread usability as state-of-the-art architectures such as transformers (Vaswani et al., 2017) consume considerable computational resources and energy (Strubell et al., 2019), release undesirable pollutants such as greenhouse gases, and is less deployable in a production environment, the latter contradicting the task's objective of enabling widespread and rapid processing of legal documents.

A system is considered to be lightweight by the authors' definition if it is capable of completing its training in a maximum of several hours on a single machine without specialized hardware, such as GPUs or TPUs. This is the case for CRFs and systems employing pre-trained models.

With regard to interpretability, the aim is to provide a means for end-users (*i.e.*, legal profession-

[1] Given that the output of the L-NER system was used as input in some RR systems, the paper will first present work done on the L-NER sub-task and subsequently the RR sub-task.

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 1412–1420

als) to query the system on how a specific decision was derived. Exposing the workings of the system would build trust for users if its explanations are deemed acceptable by the experts. In the event where they challenge the system's predictions as they are found to not be logical, the feedback received that can be used to identify and target specific system mechanics involved would also boost the robustness of the system.

Explainability is especially characteristic of CRFs, while more complicated systems such as Deep Learning approaches that employ standard real-valued tensors (*i.e.*, $x \in \mathbb{R}^{d_1 \times d_2 \times \ldots \times d_n}$) are not as interpretable.

The authors' participation was an attempt to test the capabilities of lightweight systems in achieving reasonable performance on specialized data, given a lack of no prior knowledge or competencies in the legal domain that could provide leverage in tailoring systems for legal processing.

The code is available in a GitHub repository (https://github.com/LouisEsteve/LegalEval2023).

The rest of the paper is structured as follows: (1) Presentation of the two sub-tasks (Section 2); (2) Systems overview in (Section 3); (3) Experimental setup (Section 4); (4) Results and error analysis (Section 5); (5) Conclusion and way forward (Section 6).

2 Background

The datasets used in both sub-tasks and the division into TRAIN, DEV and TEST sets were provided by the LegalEval organizing team as JSON files.

2.1 L-NER

The L-NER sub-task aims to perform Named Entity Recognition in the legal domain, a practical application being, for example, the automated retrieval of relevant information. To do so, thirteen classes were to be annotated (details in Table 1). Of note are five classes relating to people, which will be relevant for analysis in later sections. The datasets for this sub-task, as shown in Table 2, are themselves split into preamble and judgement; both were used interchangeably to train and estimate systems' performance. It should also be noted that the preamble seems to contain significantly more entities relative to its number of sentences than the judgement.

Grishman and Sundheim (1996) were the first to discuss "named entity recognition" (NER) in the context of the MUC conferences. They then tackled rather general entity types (person, organization, location) which since have become standard, but focus has now shifted toward NER applied to specialized fields, such as legal documents. Specialized fields, as their name suggests, are prone to specific vocabularies and/or language structures, making it harder to process using systems and techniques from general domain.

The literature contains some references of NER specific to the legal domain : Dozier et al. (2010) explore symbolic approaches (*i.e.*, "lookup", "contextual rules", "statistical models") achieving high precision (0.84-0.98) and good recall (0.72-0.87), while Vardhan et al. (2021) use Convolutional Neural Networks (CNNs) and achieve a precision of approximately 0.73 and a recall of 0.50. It should

Entity Type	**Description**
CASE_NUMBER	Case number
COURT	Court
DATE	Date
GPE	Geopolitical entity
JUDGE	Judge
ORG	Organization
OTHER_PERSON	Other person (*i.e.*, a person neither PETITIONER, RESPONDENT, JUDGE, nor WITNESS)
PETITIONER	Petitioner (*i.e.*, person initiating legal process)
PRECEDENT	Precedent (*i.e.*, former legal decisions)
PROVISION	Provision
RESPONDENT	Respondent (*i.e.*, person facing legal process)
STATUTE	Statute
WITNESS	Witness

Table 1: Classes in the L-NER sub-task

Sub-corpus	**# Texts**	**# Entities**
TRAIN preamble	1560	12479
TRAIN judgement	9435	17485
DEV preamble	125	1385
DEV judgement	949	1876
TEST	4501	-

Table 2: Description of the datasets of the L-NER sub-task. (Number of texts = number of preambles or judgement sentences)

Rhetorical Role	Description
PREAMBLE	Preamble
FAC	Facts
RLC	Ruling by Lower Court
ISSUE	Issues
ARG_PETITIONER	Argument by petitioner
ARG_RESPONDENT	Argument by respondent
ANALYSIS	Analysis
STA	Statute
PRE_RELIED	Precedent relied
PRE_NOT_RELIED	Precedent not relied
RATIO	Ratio of the decision
RPC	Ruling by Present Court
NONE	-

Table 3: Classes in the RR sub-task

Sub-corpus	# Documents	# Sentences
TRAIN	244	28986
DEV	30	2890
TEST	50	4158

Table 4: Description of the datasets of the RR sub-task

be noted that these two references do not perform the same task, and they do not use the same data sets, therefore a strict comparison may not be entirely justifiable.

For a broader view of different architectures on a variety of specialized domains, refer to Liu et al. (2021); upon testing multiple architectures, they come to the conclusion that BERT (Devlin et al., 2019) architectures perform well on a variety of domains. However, as previously mentioned, with lightweight systems in mind, system development has been limited to Conditional Random Fields (Lafferty et al., 2001).

2.2 RR

The RR prediction sub-task deals with text classification in Indian court judgements written in English by legal professionals and often of a substantial length (average of 119 sentences/document in the TRAIN dataset). This sub-task differs from the traditional text classification task of assigning one label to an entire document. Instead, the document is to be decomposed into rhetorical roles (RRs) and a class is to be assigned to each RR found.

12 RR classes have been identified (Table 3 [2]), some of which pertain specifically to legal documents, *e.g.* RLC (Ruling by Lower Court), ARG_PETITIONER. Another label has been added in order to classify sentences that do not belong to any of the 12 RR classes: NONE. Each sentence must be labelled with one of the 13 classes (12 informative and NONE).

Some statistics characterizing the datasets provided by the organizers can be found in Table 4.

This task has been previously explored and tested on different corpora with different sets of Rhetorical Roles. Hachey and Grover (2006) worked on the summarization of legal texts based on the classification of sentences into 7 RR classes (fact, proceedings, background, etc.). They tested 5 classifiers and obtained the best performances with SVM and maximum entropy sequence models. Saravanan and Ravindran (2010) proposed a more efficient rule-based method based on CRFs. More recently, Bhattacharya et al. (2019) tested deep learning methods based on BiLSTM. Their best system is a BiLSTM model with a CRF output layer.

3 System overview

3.1 L-NER

With an objective of reasonable interpretability, a standard CRF (Lafferty et al., 2001) was chosen for the L-NER task. The Python library `sklearn-crfsuite` [3] was used.

The overall structure is as follows:

1. Process the texts with SpaCy (Honnibal et al., 2020) and retrieve token-level features
2. For each token of each text, add as features those of previous and upcoming tokens on an $-m : +n$ span, such as `-2:head` for the syntactic head of the token two positions ago, or `+1:morph` for the morphological aspects of the token one position ahead
3. Feed this structure to a CRF
4. Apply the CRF model on the TEST set
5. Optionally, apply post-processing (regular expressions to extract dates, case numbers and geopolitical entity)

[2] https://github.com/Legal-NLP-EkStep/rhetorical-role-baseline

[3] https://pypi.org/project/sklearn-crfsuite/

The regular expressions for the post-processing step were constructed using the TRAIN set for reference. For DATE, a variety of formats were included, such as digit-based (*e.g.*, DD-MM-YYYY, MM-DD-YYYY), or equivalents with months written in letters. For CASE_NUMBER patterns, this included specific indicators (*e.g.*, "civil", "criminal", "appeal") adjacent to the token being processed. For GPE, this consisted of a very basic (and naive) pattern: a token with at least three capital letters. Several instances of tokens following this format were observed, however, deploying this regular expression proved to be detrimental.

For further information with regards to the Python version, module versions, and SpaCy model used, please refer to the GitHub repository (https://github.com/LouisEsteve/LegalEval2023).

3.2 RR

The RR task can be seen as a sequence labeling task or as a sentence classification task. The authors have opted for the second way of conceptualizing this process: the document is first divided into sentences, and each sentence unit is then assigned a label. An RR block is then rebuilt from joining contiguous sentence units of the same class.

The task of identifying Rhetorical Roles (RR) was tackled with several approaches, of which the two best-performing methods selected are: (1) a Bag-of-Words system, and (2) a sentence-transformers Deep Learning vector-based approach [4]. As an optional pre-processing step, the entities in the input sentences were first labelled with our NER system (developed for the L-NER sub-task), and then replaced by their type. This was to reduce the variability of the sentences and to enrich them with semantic information.

[Bag-of-Words] As for the Bag-of-Words system, the model was developed using the `scikit-learn` library (Pedregosa et al., 2011). Our findings indicate that the linear regression algorithm outperformed other algorithms available. To achieve this result, we used a combination of TF-IDF weights and tested out uni- and bi-grams.

[sentence-transformers] For the vectorial approach, each datapoint was converted into a sentence vector using the SBERT sentence transformer [5] (Reimers and Gurevych, 2019). The transformed data was then fed to a K-Neighbours model. The input test dataset was also converted into a sentence vector per datapoint for the model to classify into the 13 classes.

Both of these approaches classify sentences independently into one of the 13 classes. In order to move from sentence classification to text segmentation, we added a rule-based post-processing phase.

The post-processing rules are as follows:

- Search for the last sentence of the preamble (keyword: "JUDGMENT"): all the sentences before are tagged as PREAMBLE;
- Process all the following sentences:
  - If the sentence tag is PREAMBLE, change the label to the one of the previous sentence;
  - Else, if the sentence tag is one of the low-proportion classes (RATIO, ARG_PETITIONER, ARG_RESPONDENT, RLC, STA, PRE_NOT_RELIED, PRE_RELIED), continue;
  - Else, compare the label of the current sentence with the previous and next ones, if the label of the previous and the next are identical, assign this tag to the current sentence.

4 Experimental setup

The TRAIN and DEV data split correspond to the raw TRAIN and DEV datasets provided by the organizers. During training, systems were trained on TRAIN and tested on DEV, but for the final version, they were trained on both TRAIN and DEV.

4.1 L-NER

With regards to features, we have tried a variety of different configurations, arriving at the following features extracted with SpaCy: the raw token, the fine-grained PoS, the shape as in capitalization and presence of digits, named entity types detected by SpaCy, the type of syntactic dependency, the syntactic head, and morphological aspects. These features have been considered in a window of 4

[4] To compare different approaches, a transformer method was tested as it is lightweight in the sense of it being a pre-trained model, even if it is technically not lightweight if it had to be fine-tuned on legal data, and not as interpretable as the other systems developed.

[5] https://www.sbert.net/index.html

tokens before and 4 tokens after the current token. Hyperparameters can be seen in Table 5.

Different types of post-processing were tested with the aim of increasing recall of certain classes. On the DEV set, we have seen some improvement for the types DATE, CASE_NUMBER and GPE.

Algorithm	pa[6]
Number of epochs	200
c1	0.0
c2	1.0
epsilon	$1e-6$

Table 5: Hyperparameters for the CRF model

With regards to the interpretability of CRF systems, it is possible, as shown in Figure 1, to access the weight of each and every feature for all tags. For each one, there is access to its polarity (*i.e.* lower or higher than zero), and intensity (*i.e.* its distance to zero). For example, in Figure 1, the most relevant feature used in tagging a token as B-CASE_NUMBER appears to be the shape X.X.Xx.dddd/dddd (X: upper case letter, x: lower case letter, d: digit character).

4.2 RR

[Bag-of-Words] A five-fold cross-validation technique was employed, with a grid search to optimize the hyperparameters summarized in Table 6.

The system was tested with and without preprocessing (the replacement of named entities detected by the L-NER system with their type).

vectorizer	TfidfVectorizer
classifier	LogisticRegression()
classifier__C	1.0
vectorizer__max_features	5000
vectorizer__ngram_range	(1, 2)

Table 6: Hyperarameters for the Bag-of-Words system

[sentence-transformers] The sentence-transformers [7] architecture and pretrained sentence embedding model all-MiniLM-L6-v2 (Reimers and Gurevych, 2019)[8] were employed as-is to generate sentence embeddings for both

[6]Stands for "Passive Agressive" according to sklearn-crfsuite's documentation (Crammer et al., 2006).

[7]https://www.sbert.net/index.html

[8]https://huggingface.co/nreimers/MiniLM-L6-H384-uncased

y=B-CASE_NUMBER top features

Weight[?]	Feature
+0.219	shape_:X.X.Xx.dddd/dddd
+0.172	text:Crl
+0.168	head:in
+0.162	+1:shape_:Xx.ddd/dddd
+0.140	+1:text:Petition
+0.137	text:Criminal
+0.131	+1:shape_:dddd/dd
+0.130	+1:text:Appeal
+0.123	text:Cr
+0.121	+1:head:.
+0.120	text:Civil
+0.117	+1:shape_:Xx.dddd/dddd
+0.114	shape_:X.X.Xx.ddd
+0.114	head:Appeal
+0.114	-1:text:passed
	... 8865 more positive ...
	... 21529 more negative ...
-0.115	last_word:Judgment
-0.118	first_word:High
-0.120	-1:shape_:Xxx
-0.220	-1:dep_:compound
-0.237	shape_:Xxxxx

Figure 1: Interpretability of CRF systems. Example of the weight of some features for the B-CASE_NUMBER label (i.e. first token of an entity of type CASE_NUMBER) obtained using Python package eli5.

datasets without any fine-tuning due to material constraints. A scikit-learn[9] K-Neighbours classifier was then trained on the TRAIN dataset, and later employed to generate predictions on the DEV dataset. According to the test runs conducted, the best results across all evaluation metrics were obtained with $k = 15$. We have also experimented the use of the output of the L-NER system in order to substitute entities by their type before to generate sentence embeddings.

The rules-based post-processing method was also tested on both approaches. On the DEV set, it improves the accuracy by an average of 6 points for the BOW system.

5 Results

In this section, results on the DEV dataset and evaluation results on the TEST dataset (as published on the codalab submission platform) are listed. An error analysis is also performed on each system.

[9]https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

5.1 L-NER

Results on the DEV set and results obtained upon evaluation on the TEST set are listed in tables 7 and 8. Post-processing did increase recall, but the detriment to precision was greater. The regular expressions used for post-processing were determined in earlier steps of system design, a time at which they proved more useful than in this last system.

Class	Precision	Recall	$F_{\beta=1}$
CASE_NUMBER	0.625	0.580	0.601
CASE_NUMBER (+ post-proc.)	0.625	0.580	0.601
COURT	0.859	0.815	0.836
DATE	0.901	0.930	0.915
DATE (+ post-proc.)	0.601	0.949	0.736
GPE	0.705	0.517	0.597
JUDGE	0.907	0.880	0.893
LAWYER	0.934	0.861	0.896
ORG	0.564	0.454	0.503
ORG (+ post-proc.)	0.328	0.532	0.405
OTHER_PERSON	0.769	0.699	0.732
PETITIONER	0.741	0.642	0.688
PRECEDENT	0.702	0.728	0.715
PROVISION	0.902	0.867	0.884
RESPONDENT	0.720	0.614	0.663
STATUTE	0.882	0.878	0.880
WITNESS	0.826	0.678	0.745
Macro average (no post-proc.)	0.788	0.725	0.753
Macro average (with post-proc.)	0.750	0.731	0.734

Table 7: Results on DEV (L_NER_CRF_model_101)

Post-processing	Macro F1
No post-processing	**0.649**
DATE	0.641
CASE_NUMBER	0.624
GPE	0.597
DATE, CASE_NUMBER, GPE	0.526

Table 8: Results upon evaluation on TEST

The best results were obtained on the TEST set without post-processing. The baseline system proposed by the organizers (Kalamkar et al., 2022a) obtained a Macro $F_{\beta=1}$ of 0.911 and the team ResearchTeam_HCN, ranked first at the LegalEval 2023 shared task, reached a Macro $F_{\beta=1}$ of 0.912 (Modi et al., 2023). These two systems were based on RoBERTa-base.

Results on the TEST set (Macro $F_{\beta=1} = 0.64$) were lower than on the DEV set (Macro $F_{\beta=1} = 0.74$). Having no access to the TEST set at the time of writing, it is not possible to accurately assess the reasons behind such a difference in performance. Under such circumstances, the default hypothesis would be that our system could not generalize well enough for its performance to be consistent over different data sets.

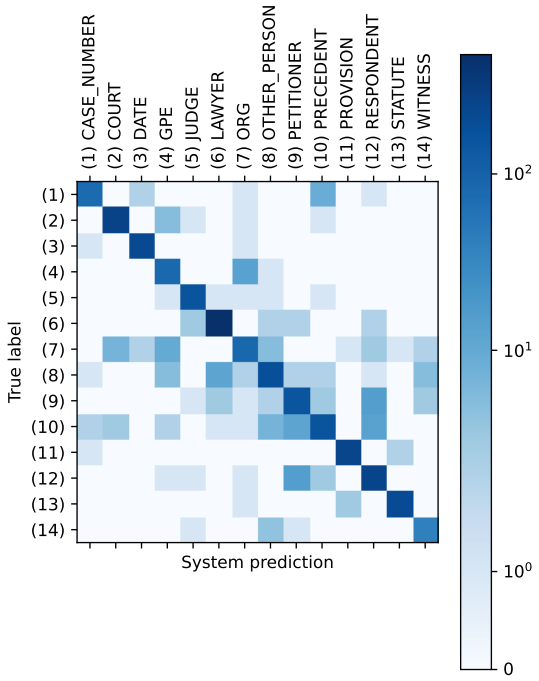

Figure 2: Confusion matrix on the DEV set

Regarding errors, analysis of the version trained on TRAIN and tested on DEV shows that the system's main issue lies between RESPONDENT and PETITIONER; overall, the frontiers match the reference but the two classes tend to be mistakenly annotated as the other. A more exhaustive confusion matrix is available in Figure 2.

A meta-observation made is that it is entities of the general type of people that causes difficulties for the system, implying that differentiating the specific roles of people goes beyond the CRF system's capabilities.

The effect of the post-processing phase can be observed from row 2 and onwards in Table 8. DATE, CASE_NUMBER, and GPE were all post-processed using regular expressions based on what we could extract from the TRAIN set. It seems that basing regular expressions on limited data does not ensure high quality, as the differences in score suggest. Even for DATE that was thought to have reasonably stable patterns (*e.g.* ISO-8601), ground truth indicates that strict patterns do not seem to be a reliable occurrence.

5.2 RR

Results of the systems on DEV and results obtained on the TEST set for the RR sub-task are as follow in tables 9 and 10:

Model	Macro F1
Bag-of-Words	0.37
Bag-of-Words + PP	0.39
Bag-of-Words + NER	0.39
Bag-of-Words + NER + PP	**0.41**
sentence-transformers	0.35
sentence-transformers + PP	0.37
sentence-transformers + NER	0.35
sentence-transformers + NER + PP	0.37

Table 9: Results on the DEV set (PP = post-processing)

Model	Macro F1
Bag-of-Words + NER	**0.568**
sent-transformers	0.552
sent-transformers + NER	0.551

Table 10: Results upon evaluation on TEST set (system with post-processing)

As shown in Table 9, the best results are obtained by the Bag-of-Words system with pre- and post-processing phases. Substitution of the entities detected by the L-NER system improves the performance of the Bag-of-Words system by two points, but does not impact the results of the sentence-transformers system (slight increase in precision).

There is a significant increase in the macro $F_{\beta=1}$ metric from the DEV to TEST results. One hypothesis is that the two datasets are rather varied but the systems are able to generalize across the two datasets, performing even better on the TEST dataset. On the TEST set, the Bag-of-Words system that utilized pre-annotated data from the L-NER task achieved the best results (Macro $F_{\beta=1} = 0.568$). The other approach trialed, sentence-transformers, followed close behind (Macro $F_{\beta=1} = 0.551$). The sentence-transformers method was also tested with the pre-processing lemmatisation step, obtaining poorer results (Macro $F_{\beta=1} = 0.529$) than without. The system that ranked first at the LegalEval 2023 shared task, developed by the AntContentTech team, obtained a Macro $F_{\beta=1}$ of 0.859. It is based on BiLSTM and CRF, and employed the LegalBERT model.

An error analysis of the systems tested on the DEV dataset showed that unsurprisingly, the best-represented classes (`ANALYSIS`, `FAC` and `PREAMBLE` making up 2,076 out of the 2,890 sentences) tended to perform the best across all 3 metrics (precision, recall and F1) among the 13 classes, while the least-represented classes (`PRE_NOT_RELIED`) performed the worst in all 3 metrics or obtained very lopsided results (1.0 in precision and 0.0 in recall and F1).

The most represented class, `ANALYSIS`, obtained the best recall across systems but an average score in precision. This signifies while the systems did well in identifying positives, this was done in excess and generated many false positives. This includes a tendency to confuse between `ANALYSIS` and `FAC`. This may have been an unavoidable consequence due to the dominance of this class, given that `ANALYSIS` alone accounts for 34% of the total number of sentences for all 13 classes.

As for the least-represented classes, one way to improve the systems' performance is to increase the number of samples in the corpus to level that of other classes so that the systems could generalize across more sentences and combat the dominance of one or a few classes.

6 Conclusion

Our participation in the LegalEval task was motivated by the express aim of evaluating the competitiveness of lightweight systems, with the interpretability of the system as a bonus criterion. Results obtained in the RR and L-NER tasks were modest, at 0.568 and 0.649 and placing 25th (out of 27) and 14th (out of 17) respectively.

For the L-NER task, we found that (1) CRFs may attain a reasonable performance level (Macro $F_{\beta=1} \approx 0.74$ on DEV) when trained to annotate all the classes at once, but (2) they face issues when having to differentiate between very similar classes (*e.g.*, between `PETITIONER` and `RESPONDENT`) and (3) it is not certain if they are really capable of generalizing across datasets (Macro $F_{\beta=1} \approx 0.64$ on TEST, 10 points below DEV). For the RR task, we found that replacing named entity spans with their class slightly improves performance of RR systems to a certain extent. The synergy between the two sub-tasks should be capitalized on by deploying the systems in that order.

With regard to the way forward, the L-NER system could potentially be improved in terms of its

raw performance by incorporating an LSTM before the CRF, as shallow (and with rather few cells in width) LSTMs are considered to boost performance (Reimers and Gurevych, 2017) while keeping the model lightweight. This may go towards improving the detection of rhetorical roles for the BOW RR system as well. However, this could possibly degrade both the highly-human interpretable NER and BOW RR systems' interpretability. As for the sentence-transformers RR system, it could be considered 'lightweight' in its current form that makes use of an existing trained model available as a library. However, the obvious method of improving the system by fine-tuning it on legal documents would render the system a classic Deep Learning heavyweight system.

Acknowledgments

We wish to thank the authors of the task for their work and their answers to various questions that were asked on the dedicated mailing list. The insightful and sharp advice of our reviewers was greatly appreciated, enabling us to meaningfully focus our efforts on improving the article.

References

Paheli Bhattacharya, Shounak Paul, Kripabandhu Ghosh, Saptarshi Ghosh, and Adam Wyner. 2019. Identification of rhetorical roles of sentences in indian legal judgments. In *Legal Knowledge and Information Systems - JURIX 2019: The Thirty-second Annual Conference, Madrid, Spain, December 11-13, 2019*, volume 322 of *Frontiers in Artificial Intelligence and Applications*, pages 3–12. IOS Press.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7(19):551–585.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Christopher Dozier, Ravikumar Kondadadi, Marc Light, Arun Vachher, Sriharsha Veeramachaneni, and Ramdev Wudali. 2010. Named Entity Recognition and Resolution in Legal Text. In Enrico Francesconi, Simonetta Montemagni, Wim Peters, and Daniela Tiscornia, editors, *Semantic Processing of Legal Texts: Where the Language of Law Meets the Law of Language*, pages 27–43. Springer Berlin Heidelberg, Berlin, Heidelberg.

Ralph Grishman and Beth Sundheim. 1996. Message Understanding Conference- 6: A Brief History. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, volume 1, pages 466–471.

Ben Hachey and Claire Grover. 2006. Extractive summarisation of legal texts. *Artif. Intell. Law*, 14(4):305–345.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python. Original-date: 2014-07-03T15:15:40Z.

Prathamesh Kalamkar, Astha Agarwal, Aman Tiwari, Smita Gupta, Saurabh Karn, and Vivek Raghavan. 2022a. Named entity recognition in Indian court judgments. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 184–193, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

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

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Zihan Liu, Yan Xu, Tiezheng Yu, Wenliang Dai, Ziwei Ji, Samuel Cahyawijaya, Andrea Madotto, and Pascale Fung. 2021. CrossNER: Evaluating Cross-Domain Named Entity Recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13452–13460. Number: 15.

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).

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher,

Matthieu Perrot, and Edouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Nils Reimers and Iryna Gurevych. 2017. Optimal Hyperparameters for Deep LSTM-Networks for Sequence Labeling Tasks. ArXiv:1707.06799 [cs].

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Murali Saravanan and Balaraman Ravindran. 2010. Identification of rhetorical roles for segmentation and summarization of a legal judgment. *Artif. Intell. Law*, 18(1):45–76.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 3645–3650. Association for Computational Linguistics.

Harsh Vardhan, Nitish Surana, and B. K. Tripathy. 2021. Named-Entity Recognition for Legal Documents. In *Advanced Machine Learning Technologies and Applications*, pages 469–479, Singapore. Springer Singapore.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.