DeBERTa Beats Behemoths: A Comparative Analysis of Fine-Tuning, Prompting, and PEFT Approaches on LegalLensNER

Hanh Thi Hong Tran*[1,2,3], Nishan Chatterjee*[1,2,3], Senja Pollak[2], Antoine Doucet[1]
[1] La Rochelle University, France
[2] Jožef Stefan Institute, Slovenia
[3] Jožef Stefan International Postgraduate School, Slovenia
*Equal contribution
Correspondence: thi.tran, nchatter@univ-lr.fr

Abstract

This paper summarizes the participation of our team (Flawless Lawgic) in the legal named entity recognition (L-NER) task at *LegalLens 2024: Detecting Legal Violations*. Given possible unstructured texts (e.g., online media texts), we aim to identify legal violations by extracting legal entities such as "*violation*", "*violation by*", "*violation on*", and "*law*". This system-description paper discusses our approaches to address the task, empirically highlighting the performances of fine-tuning models from the Transformers family (e.g., `RoBERTa` and `DeBERTa`) against open-sourced LLMs (e.g., `Llama`, `Mistral`) with different tuning settings (e.g., `LoRA`, Supervised Fine-Tuning (`SFT`) and prompting strategies). Our best results, with a weighted F1 of 0.705 on the test set, show a 30 percentage points increase in F1 compared to the baseline and rank 2 on the leaderboard, leaving a marginal gap of only 0.4 percentage points lower than the top solution. Our solutions are available at @honghanhh/lner.

1 Introduction

The internet has revolutionized how we share and interact with information. Every day, we generate an enormous quantity of digital textual data in the form of news articles, blogs, and social media posts. The information we consume and produce, not to mention the platforms we interact on contain a multitude of legal claims, and violations are no exceptions. It is undeniable that these violations pose potential risks to individuals and organizations as well as go against the fabric of legal and ethical standards, including individual rights, societal norms, and the principles of justice.

Previous studies often trace the legal violations from their data trails by using specialized models tailored for specific domain applications (Silva et al., 2020; Yu et al., 2020). While these models can be effective in their narrow domains, they often lack the necessary versatility to address the wide array of legal violations across contexts. To address this, Bernsohn et al. (2024) formulate a new task of automatically identifying legal violations from unstructured text sources in the form of legal named entity recognition (L-NER). While baseline methods have been created to address this task, there remains a gap in developing more advanced methods to sort through this online noise and identify these breaches.

Inspired by the work of Bernsohn et al. (2024) on *LegalLens* consisting of a novel textual dataset for legal violation identification using large-scale language models (LLMs), we address a comparative analysis of different approaches on this dataset through the *LegalLens 2024: Detecting Legal Violations* shared task (Hagag et al., 2024). The contributions of this paper are two-fold:

- We propose a comparative evaluation of different techniques, including the adaptation of various language models (e.g., `RoBERTa` (Liu et al., 2019), `DeBERTa` (He et al., 2021)) as fined-tuning token classifiers against open-sourced LLMs with token classification and supervised fine-tuning using `LoRA`, and zero-shot prompt engineering approaches, gaining valuable insights into their applicability and limitations in the context of legal NLP.
- Our code is publicly available as an open-sourced repository on GitHub and our models are accessible via HuggingFace, making our work more transparent and reproducible.

The paper is organized as below. Section 2 summarizes the previous works for the L-NER task. Section 3 describes the architecture, dataset, and evaluation metrics for the comparative analysis. In Section 4, we report the performances of our methods on the development set. We also compare our best classifier on the development set with the test set against the baseline. Finally, we propose error

Proceedings of the Natural Legal Language Processing Workshop 2024, pages 371–380
November 16, 2024 ©2024 Association for Computational Linguistics

analysis in Section 5, followed by the conclusion with future works in Section 6.

2 Related Works

The primary works for legal violation identification were mostly on domain-specific topics such as data agreements for compliance (Amaral et al., 2023), data privacy breaches (Silva et al., 2020), and industry-specific regulations (Nyffenegger et al., 2024; Yu et al., 2020). Despite their potential, these studies suffered from the limitation of specific types of legal domains or particular sectors.

One of the most popular directions for legal violation identification was to consider the task as a named entity recognition (Hanh et al., 2021; Ivačič et al., 2023; González-Gallardo et al., 2024) task, or so-called L-NER. In non-neural approaches, Dozier et al. (2010) extracted the named entities (NEs) in the US case law and many other legal documents by implementing list lookups, contextual rules, and statistical models. In neural ones, Leitner et al. (2019) suggested a biLSTM-CRF model for their novel manually annotated datasets about German court decisions with 19 NEs while others proposed LSTM-CRF for LeNER-Br[1] legal documents in Brazilian. Chalkidis et al. (2020) presented LEGAL-BERT[2] with different BERT-based model fine-tuned on 12 GB of English legal texts. Further works (Vardhan et al., 2021) elaborated the neural architecture for legal identification via NER task by convolutional neural networks (CNN) and multi-layer perceptions (MLP). Several other language models (e.g., BERT, DistilBERT, RoBERTa) were also fine-tuned to enhance the performance of legal violation identification (Bernsohn et al., 2024) in the same LegalLens[3] corpora.

With the advent of large-scale language models (LLMs), numerous works have been done to take advantage of LLMs to [1] explain legal terms present in legislative documents (Nyffenegger et al., 2024), [2] analyze the legal textual data (e.g., court decision analysis, rivalling seasoned law students) in depth (Savelka et al., 2023), [3] generate synthetic data in legal domains (Oliveira et al., 2024; Bernsohn et al., 2024), or [4] fine-tune a specialized classifier (e.g., `Llama-2`) for the downstream task (Bernsohn et al., 2024), to mention a few.

[1] https://github.com/peluz/lener-br
[2] https://github.com/nonameemnlp2020/legalBERT
[3] https://github.com/darrow-labs/LegalLens

3 Methods

In this section, we explore three different setups to tackle the challenge of the L-NER task, including: [1] We evaluate Transformers variants (e.g., `RoBERTa` (Liu et al., 2019), `DeBERTa` (He et al., 2021), and `DeBERTa-LSTM`) through the process of fine-tuning; [2] We explore prompting LLMs in zero-shot settings (Li, 2023) with different fine-tuned checkpoints (e.g., `Mistral` (Jiang et al., 2023), `Llama-2` (Touvron et al., 2023a), `Llama-3.1` (Dubey et al., 2024)); and [3] We perform parameter-efficient fine-tuning (`PEFT`) using low-rank adaptation (`LoRA`) with LLMs.

3.1 Architecture

Fine-tuning on the Transformers family: We evaluate the effectiveness of transformer-based language models by fine-tuning `RoBERTa`[4] (as a baseline) and `DeBERTa`[5] with and without an additional `LSTM` layer (Hochreiter and Schmidhuber, 1997) following the success of Bernsohn et al. (2024). We train the models using the AutoModel classes from the HuggingFace Transformers library. Each model was trained for 10 epochs with an initial learning rate of $2e-5$, batch size of 16, warm-up steps of 500, weight decay of 0.01, random seed of 42, and a max sequence length of 512 tokens. For the additional layers incorporating `DeBERTa`, we set the dropout rate to 0.3. Early stopping was applied to prevent overfitting.

Prompting LLMs in Zero-Shot Settings: We evaluate several open-sourced instruction-tuned LLMs to test their ability on this task. In zero-shot settings, we treat the L-NER task as a slot-filling problem, where each slot corresponds to a class label. We use three different prompts, where: [1] Prompt 1 is similar to the implicit prompt Bernsohn et al. (2024) used for their few-shot classification setting; [2] Prompt 2 is what Bernsohn et al. (2024) used to create their dataset using GPT-4 (OpenAI et al., 2024) before human annotation; and [3] Prompt 3 is based on rephrasing the prompt explicitly as a slot-filling problem instead of a NER task. The prompts can be seen in Figure 2. We use the JSONFormer[6] to constrain the outputs into a structured format. The top experiment's results

[4] https://huggingface.co/FacebookAI/roberta-base
[5] https://huggingface.co/microsoft/deberta-v3-base
[6] https://github.com/1rgs/JSONFormer

have been listed in Table 1, while the complete list can be found in Table 6 in the Appendix B. This helps us understand whether fine-tuning is necessary for tackling this task and identify potential candidates for fine-tuning.

LoRA with Open-Sourced LLMs: We experiment using different open-sourced LLM families, including `Qwen2` (Yang et al., 2024), `Mistral` (Jiang et al., 2023), `Llama-2` (Touvron et al., 2023b), and `Llama-3x` (Dubey et al., 2024). We consider the same data sizes of 7-8 billion parametric versions for all the tested LLMs. Following the success of `PEFT` for fine-tuning LLMs as a token classifier, we leverage `LoRA` (Hu et al.), a fine-tuning technique that adds a small, low-rank matrix to the pre-trained model weights, allowing for efficient adaptation to new tasks with fewer trainable parameters. `LoRA` works by keeping the majority of the model's weights frozen and only training a small number of parameters specific to the task, drastically reducing the computational cost while maintaining high performance. Each model was trained for the same 10 epochs with a batch size of `LoRA r` of 12, `LoRA alpha` of 32, and `LoRA dropout` of 0.1. We use Li et al. (2023)'s `LlamaForTokenClassification` and `MistralForTokenClassification`, which use Label Supervision (`LS`) to constrain the output predictions. In addition, we perform Supervised Fine-tuning (`SFT`) using `LoRA` on `Llama3.1-8b` (Dubey et al., 2024) using the `Llama-3` instruction format to produce JSONFormer-like JSON outputs. We use the same `LoRA` configurations as before for training and JSONFormer for testing.

3.2 Datasets

We use the training and development sets from LegalLens (Bernsohn et al., 2024) designed for the L-NER task to identify violations with four distinct classes: "*violation*", "*violation by*", "*violation on*", and "*law*". The class description, the number of instances per class, and their average phrase length are presented in Table 4 in Section A.

3.3 Evaluation Metrics

The L-NER task's performance is assessed using Precision, Recall, and weighted F1-score.

4 Results

Table 1 presents the performance of different models given three settings: [1] Fine-tuning (e.g.,

Table 1: Comparison of different methodologies for L-NER on the development set. The table showcases various models, their sizes, the methods employed, and their performance metrics, where P is Precision, R is Recall, and F1 is the F1-score. Both Prompting and `SFT` use Prompt 2 as the instruction (see Figure 2).

Models	Size	Methods	P	R	F1
RoBERTa	125M	Fine-tune	0.568	0.674	0.616
DeBERTa-v3	250M	Fine-tune	**0.633**	0.664	**0.648**
DeBERTa-v3+LSTM	250M	Fine-tune	0.577	**0.688**	0.627
Mistral-v0.3	7B	Prompting	0.246	0.258	0.252
Llama-2-hf	7B	Prompting	0.122	0.173	0.143
Dolphin-2.9-Llama-3	8b	Prompting	0.425	0.509	0.463
Meta-Llama3.1	8B	Prompting	0.456	0.282	0.348
Qwen2	7B	LS-LoRA	0.228	0.333	0.270
Mistral-v0.3	7B	LS-LoRA	0.160	0.272	0.202
Llama-2	7B	LS-LoRA	0.372	0.536	0.439
Dolphin-2.9-Llama-3	8B	LS-LoRA	0.228	0.370	0.282
Llama-3.1	8B	LS-LoRA	0.448	0.637	0.526
Llama-3.1	8B	SFT-LoRA	0.015	0.110	0.027

`RoBERTa`, `DeBERTa`); [2] Prompting (e.g., `Mistral`, `Llama`); and [3] `LoRA` (e.g., `Qwen2`, `Mistral`, `Llama`). In general, all the fine-tuned BERT-based language models outperform LLMs for both `LoRA` and instruction-tuning settings by a significant margin. Across all models, `DeBERTa` attains the best performances, achieving an F1 of 64.8% and a Precision of 63.3% on the development set.

Given the best performance on the development set of DeBERTa as a fine-tuned token classifier, we reported the results in weighted F1 of our classifier on the hidden test set in comparison with other competitors and the baseline from the *LegalLens 2024: Detecting Legal Violations* task in Table 2.

Table 2: Results on the test set in the leaderboard.

Teams	F1
Nowj	0.416
Flawless Lawgic (Ours)	**0.402**
UOttawa	0.402
Masala-chai	0.380
UMLaw & TechLab	0.321
Bonafide	0.305
Baseline	0.381

For the LegalLens NER part of the shared task (Hagag et al., 2024), all competitors performed higher than the baseline, where our team obtained second place with only a marginal gap of 4 percentage points from the winning solution on the test set.

5 Error Analysis

Entity Type Errors: Figure 1 visualizes the comparison in F1 performance for each class among different models reported in Table 1.

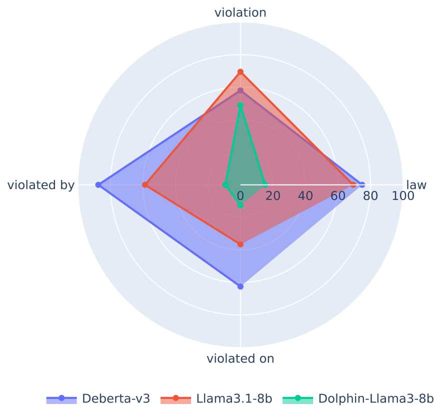

Figure 1: Comparing performance in F1 of models from Table 1 on the development set for each class.

Of all classes, the entity type "*violation*" had the lowest F1 despite its richness in training examples, especially for longer and more complex entities, followed by "*violated on*". DeBERTA showed the most competitive performance for all classes, especially in identifying the entity types "*violated on*" and "*violated by*" by a large margin. The performance of our best classifier on the development set is reported in Table 3. This indicates that training separate models for each class, or certain classes grouped together might be an interesting avenue to explore.

Additionally, the three datasets exhibit significant variability, as illustrated by the distinct class coverage of models in Figure 3 in Appendix B, which provides insights into the data distribution. This variability may explain why models trained on the training set may not generalize well to the development and test sets. Moreover, analyzing the named entities present in each slot and examining how various models perform about these, could yield additional valuable findings.

Table 3: Results per class on the development set using DeBERTA token classifier.

Classes	Precision	Recall	F1-score
LAW	0.928	0.853	0.888
VIOLATED BY	0.969	0.840	0.900
VIOLATED ON	0.608	0.600	0.604
VIOLATION	0.574	0.627	0.599

The Disparity in Performance: Although DeBERTa outperformed other masked language models of smaller size (e.g., RoBERTa), a larger model size does not always lead to better performance, especially when LoRA fine-tuning is used, which can sometimes lead to poorer results. This is consistent with the results of Li et al. (2023), which highlighted the difficulties in fine-tuning the LLMs compared to the smaller masked language models (e.g., BERT), especially when the amount of training data is limited.

Furthermore, we acknowledged the difference in objective functions between DeBERTa as a fine-tuned token classifier and other LLMs (e.g., Llama-3.1) as a SFT-LoRA classifier. While DeBERTa employed the per-token cross-entropy objective function, LLMs fine-tuned via causal language modelling, wherein the task is to learn the joint probability distribution of all tokens by maximizing the likelihood of the data. As a result, DeBERTa provided a more fine-grained and stronger gradient signal that well constrained the class space by the number of possible entities in our dataset. This highlights the gap between masked and casual language models in token classification tasks for specific domains like L-NER. Additionally, as shown in the findings of Li et al. (2023), LS LoRA provided significant improvement over SFT-LoRA. However, there is still room for improvement when compared to DeBERTa.

Practical Use of LLMs for Legal Domain: Despite not surpassing the performance of fine-tuned and LoRA methods, prompt-based methods are still a promising tool for finding the potential violation for legal documents, especially when working with limited data of the same domain or when no annotated data is available for a given domain. While it may not be as good as models trained on dedicated annotated data (fully supervised ones), it can significantly speed up the process by suggesting the violation types later reviewed and refined by human experts.

Additionally, tools like JSONFormer, which enforce structured output constraints, can help significantly in automating these tasks. By ensuring that model outputs conform to predefined formats (e.g., JSON), these tools simplify post-processing workflows, making the outputs easier to analyze and validate using non-LLM methods, as structured formats facilitate clearer interpretation and error-checking mechanisms (Liu et al., 2024).

In-Domain Fine-Tuning: We evaluated the performance of fine-tuned DeBERTa checkpoints on several NER datasets relevant to this task [7]. Surprisingly, no significant improvement was observed compared to the base DeBERTa model. However, based on our analysis of the zero-shot performance capabilities of LLMs (see Figure 3), there appears to be greater overlap between the dataset styles of the training set and the hidden test set than between the training and development sets. This suggests that having better distributions of train-dev-test splits can help with improving upon this task. Additionally, domain-specific fine-tuning where similar patterns are reflected could also potentially enhance the performance of LLMs, although further experimentation is required to validate this hypothesis. Therefore, future work could explore fine-tuning an LLM on a legal domain corpus, which may yield better results for this and similar tasks (Jiang et al., 2024).

6 Conclusion

In this study, we presented a comparative analysis of three different approaches to identify the legal violations via the L-NER task at *LegalLens 2024: Detecting Legal Violations*, including [1] fine-tuning masked language models as token classifier; [2] zero-shot prompt engineering with LLMs; [3] fine-tuning LLMs with LoRA as token classifier. Overall, the first approach using DeBERTa as the backbone outperformed other settings, demonstrating the gap in performance between masked language models and other causal LLMs in token classification tasks, especially when the amount of training data is limited. As a result, when a complete training dataset is accessible, opting for a fully-supervised fine-tuned system remains the optimal choice. However, instruction-tuning LLMs with well-defined prompting is still a potential technique with competitive results when no annotated data is available.

Acknowledgments

The authors acknowledge the financial support from the Slovenian Research and Innovation Agency through the core research programme, Knowledge Technologies (P2-0103), the research project, Linguistic Accessibility of Social Assistance Rights in Slovenia (J5-50169), the project, Cross-lingual and cross-domain methods for Terminology Extraction and Alignment, and a bilateral project funded by the program, PROTEUS, under the grant number BI-FR/23-24-PROTEUS006. This work has also been supported by the TERMITRAD (2020-2019-8510010) project funded by the Nouvelle-Aquitaine Region, France.

[7]For example, CONLL 2003 (Tjong Kim Sang and De Meulder, 2003), OntoNotes 5.0 (Pradhan et al., 2013), and WikiANN (Rahimi et al., 2019)

References

Orlando Amaral, Sallam Abualhaija, and Lionel Briand. 2023. Ml-based compliance verification of data processing agreements against gdpr. In *2023 IEEE 31st international requirements engineering conference (RE)*, pages 53–64. IEEE.

Dor Bernsohn, Gil Semo, Yaron Vazana, Gila Hayat, Ben Hagag, Joel Niklaus, Rohit Saha, and Kyryl Truskovskyi. 2024. Legallens: Leveraging llms for legal violation identification in unstructured text. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2129–2145.

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. 2020. Legal-bert: The muppets straight out of law school. *arXiv preprint arXiv:2010.02559*.

Christopher Dozier, Ravikumar Kondadadi, Marc Light, Arun Vachher, Sriharsha Veeramachaneni, and Ramdev Wudali. 2010. *Named entity recognition and resolution in legal text*. Springer.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph

Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru,

Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Carlos-Emiliano González-Gallardo, Tran Thi Hong Hanh, Ahmed Hamdi, and Antoine Doucet. 2024. Leveraging open large language models for historical named entity recognition. In *The 28th International Conference on Theory and Practice of Digital Libraries*.

Ben Hagag, Liav Harpaz, Gil Semo, Dor Bernsohn, Rohit Saha, Pashootan Vaezipoor, Kyryl Truskovskyi, and Gerasimos Spanakis. 2024. Legallens shared task 2024: Legal violation identification in unstructured text. *Preprint*, arXiv:2410.12064.

Tran Thi Hong Hanh, Antoine Doucet, Nicolas Sidere, Jose G Moreno, and Senja Pollak. 2021. Named entity recognition architecture combining contextual and global features. In *International Conference on Asian Digital Libraries*, pages 264–276. Springer.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. *Preprint*, arXiv:2006.03654.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Nikola Ivačič, Thi Hong Hanh Tran, Boshko Koloski, Senja Pollak, and Matthew Purver. 2023. Analysis of transfer learning for named entity recognition in south-slavic languages. In *Proceedings of the 9th Workshop on Slavic Natural Language Processing 2023 (SlavicNLP 2023)*, pages 106–112.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Ting Jiang, Shaohan Huang, Shengyue Luo, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, Qi Zhang, Deqing Wang, and Fuzhen Zhuang. 2024. Improving domain adaptation through extended-text reading comprehension. *Preprint*, arXiv:2401.07284.

Elena Leitner, Georg Rehm, and Julian Moreno-Schneider. 2019. Fine-grained named entity recognition in legal documents. In *International conference on semantic systems*, pages 272–287. Springer.

Yinheng Li. 2023. A practical survey on zero-shot prompt design for in-context learning. In *Proceedings of the Conference Recent Advances in Natural Language Processing - Large Language Models for Natural Language Processings*, RANLP, page 641–647. INCOMA Ltd., Shoumen, BULGARIA.

Zongxi Li, Xianming Li, Yuzhang Liu, Haoran Xie, Jing Li, Fu-lee Wang, Qing Li, and Xiaoqin Zhong. 2023. Label supervised llama finetuning. *arXiv preprint arXiv:2310.01208*.

Yong Lin, Hangyu Lin, Wei Xiong, Shizhe Diao, Jianmeng Liu, Jipeng Zhang, Rui Pan, Haoxiang Wang, Wenbin Hu, Hanning Zhang, Hanze Dong, Renjie Pi, Han Zhao, Nan Jiang, Heng Ji, Yuan Yao, and Tong Zhang. 2024. Mitigating the alignment tax of rlhf. *Preprint*, arXiv:2309.06256.

Michael Xieyang Liu, Frederick Liu, Alexander J. Fiannaca, Terry Koo, Lucas Dixon, Michael Terry, and Carrie J. Cai. 2024. "we need structured output": Towards user-centered constraints on large language model output. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, CHI '24, page 1–9. ACM.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2023. Gpt understands, too. *Preprint*, arXiv:2103.10385.

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.

Alex Nyffenegger, Matthias Stürmer, and Joel Niklaus. 2024. Anonymity at risk? assessing re-identification capabilities of large language models in court decisions. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2433–2462.

Vitor Oliveira, Gabriel Nogueira, Thiago Faleiros, and Ricardo Marcacini. 2024. Combining prompt-based language models and weak supervision for labeling named entity recognition on legal documents. *Artificial Intelligence and Law*, pages 1–21.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke

Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152, Sofia, Bulgaria. Association for Computational Linguistics.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively multilingual transfer for NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 151–164, Florence, Italy. Association for Computational Linguistics.

Jaromir Savelka, Kevin D Ashley, Morgan A Gray, Hannes Westermann, and Huihui Xu. 2023. Can gpt-4 support analysis of textual data in tasks requiring highly specialized domain expertise? *arXiv preprint arXiv:2306.13906*.

Paulo Silva, Carolina Gonçalves, Carolina Godinho, Nuno Antunes, and Marilia Curado. 2020. Using nlp and machine learning to detect data privacy violations. In *IEEE INFOCOM 2020-IEEE conference on computer communications workshops (INFOCOM WKSHPS)*, pages 972–977. IEEE.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Harsh Vardhan, Nitish Surana, and BK Tripathy. 2021. Named-entity recognition for legal documents. In *Advanced Machine Learning Technologies and Applications: Proceedings of AMLTA 2020*, pages 469–479. Springer.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Yaoquan Yu, Yuefeng Guo, Zhiyuan Zhang, Mengshi Li, Tianyao Ji, Wenhu Tang, and Qinghua Wu. 2020. Intelligent classification and automatic annotation of violations based on neural network language model. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE.

A Dataset Statistics

We provide additional statistics and descriptions to help understand the data distribution as shown in Figure 4. The most interesting part is the distribution of data in each split: The train split has a data distribution of roughly 3:1 for `VIOLATION` to the other classes, whereas this becomes 8:1 for the development set. However, the test set has almost a 1:1 ratio. Additionally, if we look at the tokens per class, then the train and development set have comparable distributions, whereas the test set has more tokens per class.

Table 4: Entity distribution and the average length of L-NER entities in LegalLens.

Entities	# Examples			Average Length		
	Train	Dev	Test	Train	Dev	Test
`LAW`	217	75	246	8.38	3.04	19.27
`VIOLATION`	710	616	371	88.02	80.45	139.81
`VIOLATED BY`	217	75	379	5.94	2.39	16.65
`VIOLATED ON`	217	75	333	5.68	2.38	21.72

The entities include: `LAW` (specific law or regulation breached), `VIOLATION` (content describing the violation), `VIOLATED BY` (entity committing the violation), and `VIOLATED ON` (victim or affected party).

B Empirical studies on zero-shot instruction tuning

To elaborate on the potential of instruction-tuning using LLMs without the need for adequate annotated data and computation resources, we provided an ablation study on zero-shot performances to identify legal violations given 3 prompt designs where the first two prompts (P1 and P2) were inspired by the work of Bernsohn et al. (2024) and the last prompt (P3) considers the task as a slot-filling problem instead of token classification task (see the prompt examples in Figure 2).

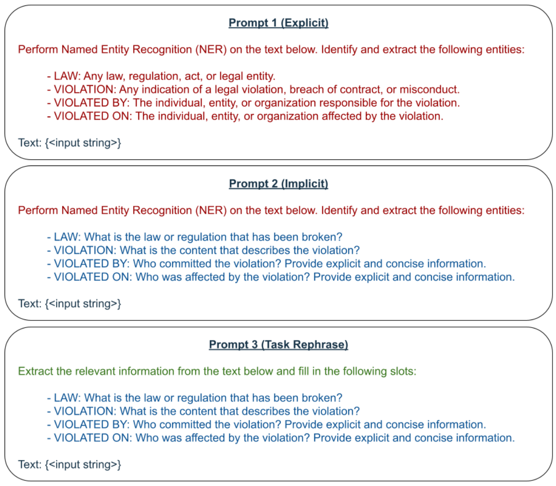

Figure 2: The three prompts we experimented with for the zero-shot setting. The color changes highlight the differences between each prompt.

Table 6 reports the zero-shot performances of three different prompt designs on the training, development, and test sets of the LegalLens dataset. Four groups of LLMs have been investigated, including [1] Llama variants (e.g., `Meta-Llama2-7b`, `Meta-Llama3-8b`, `Dolphin-Llama3-8b`, `Meta-Llama3.1-8b`); [2] Mistral variants (e.g., `Sauf-7b`, `Mistral-7b`, `Dolphin Mistral-7b`); [3] Gemma (e.g., `Gemma2-2b`); and [4] Phi (e.g., `Phi-3-mini`, `Phi-3.5-mini`). Overall, the P2 prompt structure consistently yielded better results than the other two prompts for all the tested LLMs. We suspect P2 is better because this is the style used to create the examples in the first place using GPT-4 (Bernsohn et al., 2024). Additionally, when the explicit prompts (P1) specify which items to look for, whereas P2 implicitly formulates the question.

However, using a T-test (see Table 5), we find that none of the p-values are below the common threshold of 0.05. This means there's no statistically significant difference in F1 among the three prompts. In other words, based on this test, no single prompt stands out as significantly better than the others in terms of performance. Therefore, p-tuning (Liu et al., 2023) might be an interesting dimension to explore in the future.

Table 5: T-test results for prompt comparison.

Comparison	t-statistic	p-value	Significant ($p < 0.05$)
P1 vs P2	-1.352	0.194	No
P1 vs P3	-0.366	0.718	No
P2 vs P3	1.028	0.318	No

Table 6: Zero-shot performances on the training, development, and test sets. The bold scores perform best, while the highlighted scores are models that reach over 0.4 in F1.

Model	Prompt	Train F1	Dev F1	Test F1
Saul-7b	1	0.114	0.063	0.157
	2	0.316	0.259	0.318
	3	0.259	0.171	0.266
Meta-Llama2-7b	1	0.149	0.120	0.198
	2	0.175	0.143	0.215
	3	0.152	0.110	0.177
Meta-Llama3-8b	1	0.255	0.180	0.290
	2	0.327	0.247	0.347
	3	0.294	0.195	0.322
Dolphin-Llama3-8b	1	0.406	0.334	0.422
	2	**0.463**	0.360	**0.474**
	3	0.438	**0.363**	0.451
Meta-Llama3.1-8b	1	0.254	0.195	0.305
	2	0.319	0.253	0.348
	3	0.271	0.203	0.310
Mistral-7b	1	0.166	0.082	0.262
	2	0.354	0.252	0.400
	3	0.348	0.211	0.383
Dolphin Mistral-7b	1	0.330	0.270	0.390
	2	0.424	0.356	0.419
	3	0.381	0.301	0.416
Gemma2-2b	1	0.232	0.192	0.237
	2	0.292	0.217	0.318
	3	0.182	0.146	0.199
Phi-3-mini	1	0.386	0.308	0.430
	2	0.398	0.338	0.416
	3	0.305	0.225	0.374
Phi-3.5-mini	1	0.417	0.342	0.467
	2	0.420	0.338	0.470
	3	0.377	0.287	0.425

The graph highlights significant variability across the three datasets, as evidenced by the three distinct regions, which offers valuable insights into the data distribution from a qualitative standpoint (see Figure 3). This, along with the token distribution variability as discussed in Section A helps us understand why models trained on the training set struggle to generalize effectively to the development and test sets. To further explore this, it would be beneficial to evaluate the model coverage on the other solutions across the three dataset splits.

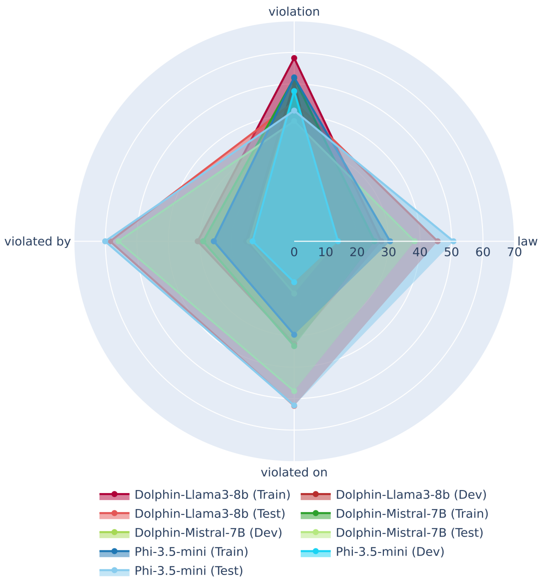

Figure 3: Per-class performance of the three models (based on overall F1) for the training, development, and test sets using zero-shot prompting. We use Prompt 2 for all since it consistently worked better than the other two across all models. Fine-grained values have been mentioned in Table 6.

It should be noted that given the token distribution, smaller LLM (up to 8b parameters as we tested) could come with the limitation of not being able to reproduce longer phrases (especially for "*violation*") which could be improved by scaling up the model sizes, especially given that the original dataset was curated using GPT-4 (Bernsohn et al., 2024).

We also find that Dolphin, the uncensored checkpoints of both `Llama-3-8b` and `Mistral-7b`, significantly outperform their aligned counterparts in the zero-shot classification task. This could be due to the alignment tax (Lin et al., 2024). However, additional qualitative investigation into the data is required before this can be confirmed.