# PositionalLJP: Multi-defendant Legal Judgment Prediction via Positional Encoding

Anonymous Author(s)

## Abstract

Legal judgment prediction (LJP) is a fundamental task in legal intelligence. While general legal judgment prediction tasks focus solely on predicting the judgment for a single defendant, multi-defendant legal judgment prediction tasks emphasize making distinct predictions for multiple defendants. Since legal documents typically present the facts in chronological order, the details relating to each defendant are intermixed. Consequently, a major challenge in multi-defendant legal judgment prediction is how to separate the facts by defendant. To address this issue, we propose a new framework called PositionalLJP, which achieves multi-defendant legal judgment prediction through positional encoding and prompt representation. Specifically, on the one hand, we construct three semantic-level prompts, each representing facts relevant to the current defendant, in order to distinguish between facts related to different defendants; on the other hand, we applied positional encoding on the occurrence positions of each defendant's name, thereby incorporating modelling of the occurrence positions and frequencies of the defendants' names. Experiments verify that our proposed method outperforms the official baseline of the CAIL2024-DRDZ competition by 2.13%. Our data and code have been released at https://anonymous.4open.science/r/PositionalLJP-CBF0/.

## 1 Introduction

With the continued increase in the number of cases and the growing complexity of the law, judicial resources have become strained. Legal Judgment Prediction (LJP) is a fundamental task of legal intelligence, involving three sub-tasks: law articles prediction, charges prediction, and terms of penalty prediction, which aims at helping people resolve legal difficulties [17]. In the first three quarters of 2026, China's people's courts accepted 727,300 criminal cases at first instance and handed down final judgments against 955,700 defendants[1], revealing that a significant proportion of these cases involved multiple defendants.

The multiple defendant prediction involves a larger number of defendants, and the prediction results for different defendants may vary. Figure 1 presents an example of multi-defendant legal judgment prediction. The case facts include the descriptions of defendants A, B, and C. Although the judgments of the law articles and charges are the same, the terms of penalty for defendants A, B and C are different. This is because A and B committed the crime again within 5 years, and due to the different previous judgment situations of A and B, the terms of penalty are also different.

The main challenge in predicting multi-defendant legal judgments is separating facts. Legal documents do not follow a sequential pattern in which each defendant describes the facts separately. Instead, facts are presented in chronological order, with details about each defendant interspersed throughout, not separated [4].

[1]https://www.court.gov.cn/zixun/xiangqing/514471.html

**Fact**: On the afternoon of February 11, 2015, around 3 p.m., defendants A, B and C conspired to steal cash by having one of them ask the victim for special-numbered 100-yuan banknotes and the other two distract the victim. They stole a total of 2,600 yuan in cash. Defendant A was previously sentenced to five years in prison and was released on January 23, 2013. Defendant B was previously sentenced to eight months in prison and was released on June 30, 2013.
**Defendants:** ['A', 'B', 'C']

| | Defendant A | Defendant B | Defendant C |
|---|---|---|---|
| Law Articles | Article 264 | Article 264 | Article 264 |
| Charges | Crime of theft | Crime of theft | Crime of theft |
| Terms of Penalty | imprisonment 8 month | imprisonment 7 month | detention 6 month |

**Figure 1: An example of multi-defendant legal judgment prediction. The case facts include the descriptions of defendants A, B, and C. Although the judgments of the law articles and charges are the same, the terms of penalty for defendants A, B and C are different.**

To address these challenges, we propose a multi-defendant legal judgment prediction framework called PositionalLJP. Specifically, we constructed three semantic-level prompts to represent the facts of different defendants and perform positional encoding on the positions where the defendant's name appears in order to achieve the effect of splitting the facts.

## 2 Related works

**Legal judgment prediction.** In recent years, most research of legal judgment prediction has focused primarily on single-defendant cases, as multi-defendant cases are generally skipped due to their complexity. Researchers focus on improving accuracy and interpretability by extracting key elements to stimulate practical judicial process [15] or injecting legal knowledge into networks [2, 5].

Research on multi-defendant LJP is still in its early stages. MAMD [7] introduced a multi-scale attention model for charge prediction in multi-defendant cases. MUD [11] introduced a new benchmark for cases involving multiple defendants, incorporating crime elements and legal rules for charge prediction. However, both MAMD and MUD are limited to charge prediction and do not address tasks such as law article prediction and penalty prediction, which to some extent limits its generalizability in LJP tasks. HRN [6] followed hierarchical reasoning chains to determine criminal relationships, sentencing circumstances, law articles, charges, and penalties for each defendant and developed a multi-defendant LJP dataset called MultiLJP. CMDL [4] is another multi-defendant LJP dataset, highlighting that existing state-of-the-art methods struggle with cases involving multiple defendants. MAGLJP [14] is a novel multi-agent framework with legal event logic graph for multi-defendant LJP.

**Positional Encoding.** The Transformer introduced Positional Encoding [10], which encodes token positional information into

the model, resulting in different representations for the same token depending on its position. In the legal field, MAMD [7] represents the positional information of defendant names as a Gaussian distribution and incorporates it into the model.

**Prompt Representation.** Advances in research on large language models (LLMs) have driven innovations in the use of prompts. Prompt learning integrates prompts into questions and answers to make it easier for models to learn [9]. Prompt formatting explores how different prompt formats and writing styles affect the effectiveness of the method [3]. Prompt representation refers to using prompts to constrain the encoder to generate text representations with specific semantic meanings [8].

## 3 Problem formulation

The legal judgment prediction includes predicting law article, charge and terms of penalty. Formally, given the fact description of a multi-defendant case as a word sequence $s = \{w_1, \ldots, w_l\}$ where $l$ represents the number of words, and a set of defendant names $D = \{d_1, d_2, \ldots, d_n\}$, where $n$ is the total number of defendants ( $n \geq 2$ ) and each name is a sequence of words $s_{d_i} = \left\{w_{d_i,1}, w_{d_i,2}, \ldots, w_{d_i,l_d}\right\}$, the task is to predict law articles $Y_a^{d_i}$, charges $Y_c^{d_i}$ and corresponding terms of penalty $Y_t^{d_i}$ for each defendant $d_i$.

## 4 PositionalLJP

In this section, we introduce the framework of proposed PositionalLJP. We use the prescribed prompt to distinguish the semantic levels of each factual description of the defendant. Specifically, we first represent the facts as origin fact, base fact, all fact and other fact through the prompt, second integrate them at the semantic level, then positional encode the positions of each defendants' name to distinguish between different defendants, and finally predict the law articles, charges and terms of penalty. The Figure 2 illustrates the flowchart of our method.

### 4.1 Fact Encoding

*4.1.1 Base fact.* Base fact represents the fact description of each defendant, and separates fact descriptions of all defendants. We can obtain the base fact description $s_{base}^{d_i}$ of each defendant from the following prompt:

> **Prompt:** Predicting {target defendant $d_i$}. {fact $s$}.

Then, we obtained the representation vector $e_{base}^{d_i}$ of the base fact through the encoder:

$$e_{base}^{d_i} = Encoder(s_{base}^{d_i}). \tag{1}$$

In this paper, we use the pre-trained model BERT [1] as the Encoder.

*4.1.2 All fact.* All fact is presented in the form of prompt and represents all the factual descriptions of the defendants. We can obtain the description $s_{all}$ of all fact using prompt:

> **Prompt:** Predicting {defendants $d_1, d_2, \ldots, d_n$}. {fact $s$}.

In the same multi-defendant case, all fact descriptions of all the defendants are exactly the same (i.e., $s_{all}^{d_i} = s_{all}$).

Similarly, we obtained the representation vector $e_{all}^{d_i}$ of the all fact through the encoder:

$$e_{all}^{d_i} = Encoder(s_{all}^{d_i}). \tag{2}$$

| Dataset | CAIL2024-DRDZ |
|---|---|
| # Training Set Cases | 12,000 |
| # Validation Set Cases | 1500 |
| # Testing Set Cases | 1500 |
| # Law articles | 31 |
| # Charges | 30 |
| # Terms of Penalty | 14 |
| # Total defendants | 48,815 |
| # Avg defendants | 3.25 |

**Table 1: Statistics of CAIL2024-DRDZ.**

*4.1.3 Other fact.* Other fact refers to the description of the facts of the defendants other than the target defendant. We can obtain the description $s_{other}^{d_i}$ of other fact from the following prompt:

> **Prompt:** Predicting {defendants $d_1, d_2, \ldots, d_n$, except target defendant $d_i$}. {fact $s$}.

Similarly, we obtained the representation vector $e_{other}$ of the other fact through the encoder:

$$e_{other}^{d_i} = Encoder(s_{other}^{d_i}). \tag{3}$$

*4.1.4 Prompt Representing.* **Strategy 1.** The base fact is presented in the form of a prompt and represents the factual description of the target defendant. It can be directly used for the representation of the facts of the target defendant. i.e. $e_{base}^{d_i}$.

**Strategy 2.** $s$ originally represent the factual descriptions of all the defendants. Firstly, we obtained the representation vector $e_{origin}^{d_i}$ of the original fact through the encoder:

$$e_{origin}^{d_i} = Encoder(s). \tag{4}$$

Then, we subtracted the facts of the other defendants from the original facts of the target defendant, and obtained the fact of the target defendant. i.e. $e_{origin}^{d_i} - e_{other}^{d_i}$.

**Strategy 3.** We subtracted the facts of the other defendants from all the facts of the target defendant, thereby obtaining the fact description of the target defendant. i.e. $e_{all}^{d_i} - e_{other}^{d_i}$.

**Fusion.** Each strategy represents a fact description of the target defendant. We combine the representations obtained from the above three methods of indicating the facts of the target defendant, in order to integrate the representations of the three strategies:

$$e_{fact}^{d_i} = W_c[e_{base}^{d_i}; e_{origin}^{d_i} - e_{other}^{d_i}; e_{all}^{d_i} - e_{other}^{d_i}] + b_c, \tag{5}$$

where ";" represents concatenate operation. $W_c, b_c$ are trainable parameters.

### 4.2 Positional Encoding

In legal judgment predictions involving multiple defendants, the names of each defendant appear at least once—and typically multiple times—in various positions. Drawing on the Transformer architecture, we apply positional encoding to the positions of the defendants' names and incorporate this into the model:

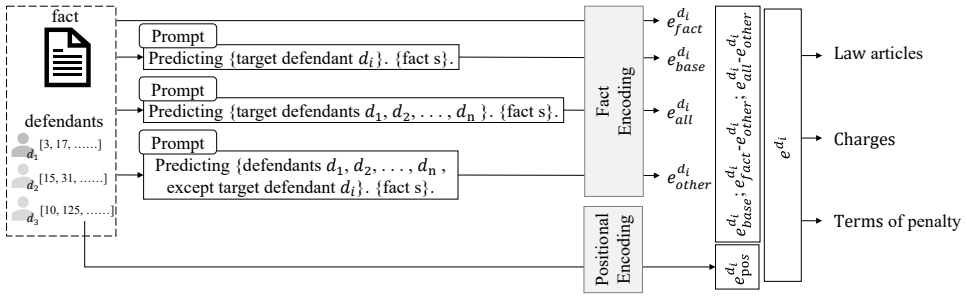



**Figure 2: The framework of PositionalLJP. (1) Represent the facts as origin fact, base fact, all fact and other fact through the prompt method.(2) Positional encode the position of defendant's name. (3) Integrate them at the semantic level.(4)Predict the law articles, charges and terms of penalty.**

| Methods | Charges | | | | Law Articles | | | | Term of Penalty | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Acc. | MP | MR | F1 | Acc. | MP | MR | F1 | Acc. | MP | MR | F1 |
| TopJudge | 54.68 | 62.91 | 52.27 | 53.06 | 57.71 | 63.48 | 55.44 | 55.75 | 24.08 | 9.92 | 11.03 | 9.24 |
| NeurJudge | 70.98 | 79.13 | 71.44 | 73.52 | 71.41 | 79.30 | 71.71 | 74.05 | 26.09 | 20.06 | 14.91 | 14.77 |
| mT5 | 74.42 | 82.46 | **80.55** | 80.70 | 74.42 | 82.27 | **80.27** | 80.44 | 23.98 | 16.53 | 13.07 | 13.43 |
| Lawformer | 70.15 | 81.12 | 75.92 | 77.90 | 70.03 | 81.50 | 75.73 | 78.00 | **29.81** | 26.57 | 21.25 | 22.73 |
| CAIL2024-DRDZ-baseline | 73.19 | **83.30** | 78.11 | 80.62 | 73.15 | 83.21 | 78.09 | 80.57 | 28.06 | 28.15 | 28.15 | 28.15 |
| MAMD | 67.32 | 79.30 | 77.81 | 78.55 | 67.45 | 79.48 | 77.94 | 78.71 | 28.32 | 28.40 | 28.40 | 28.40 |
| MUD | 74.81 | 84.75 | 78.12 | **81.30** | 74.66 | **84.51** | 78.12 | 81.19 | 29.47 | 29.33 | 29.33 | 29.33 |
| HRN-Second-level | 71.92 | 83.61 | 76.47 | 79.88 | 71.85 | 83.46 | 76.42 | 79.78 | 27.47 | 27.45 | 27.45 | 27.45 |
| MAGLJP-Conviction-agent | 71.99 | 84.53 | 74.33 | 79.10 | 71.94 | 84.28 | 74.39 | 79.03 | 28.21 | 28.05 | 28.05 | 28.05 |
| **PositionalLJP** | **75.32** | 82.71 | 79.80 | 81.23 | **75.10** | 82.67 | 79.94 | **81.28** | 27.88 | 27.88 | 27.88 | 27.88 |
| w/o Positional Encoding | 73.49 | 82.88 | 79.52 | 81.17 | 73.50 | 82.76 | 79.76 | 81.23 | **29.39** | **29.33** | **29.33** | **29.33** |
| w/o Prompt Representation | 70.48 | 82.24 | 77.99 | 80.06 | 70.28 | 82.42 | 77.87 | 80.08 | 26.67 | 26.60 | 26.60 | 26.60 |

**Table 2: Experiment result of multi-defendant legal judgment prediction on CAIL2024-DRDZ. The law Articles prediction task is conducted on article-level label set. Some of the experimental results are quoted from MAGLJP [14].**

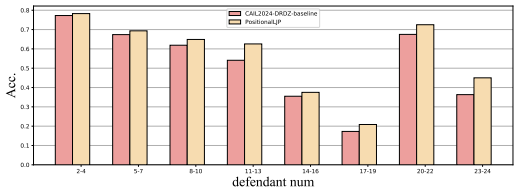



**Figure 3: Impact of the defendant number.**

$$
\begin{aligned}
PE_{(\text{pos},2i)} &= \sin\left(\text{pos}/10000^{2i/d_{\text{model}}}\right), \\
PE_{(\text{pos},2i+1)} &= \cos\left(\text{pos}/10000^{2i/d_{\text{model}}}\right), \\
e^{d_i}_{pos} &= p^{d_i} + PE(p^{d_i}),
\end{aligned} \tag{6}
$$

where pos is the position and i is the dimension. $d_{model}$ refers to the dimensions of input and output vectors in Transformer [10] and it is typically set to 512. $p^{d_i}$ is a list of the positions where the defendant's name appears in the facts of the case.

**Fusion.** Each strategy represents a fact description of the target defendant. We combine the representations obtained from the above three methods of indicating the facts of the target defendant, in order to integrate the representations of the three strategies:

$$
e^{d_i} = W[e^{d_i}_{fact}; e^{d_i}_{pos}] + b \tag{7}
$$

where ";" represents the concatenate operation.

## 4.3 Predicting and training

After obtaining the fused $e^{d_i}$, we use the following formula to predict the relevant law article, charge and term of penalty:

$$
\begin{aligned}
\hat{\boldsymbol{y}}^{d_i}_a &= \text{Sigmoid}\left(\boldsymbol{W}_a e^{d_i} + \boldsymbol{b}_a\right), \\
\hat{\boldsymbol{y}}^{d_i}_c &= \text{Sigmoid}\left(\boldsymbol{W}_c e^{d_i} + \boldsymbol{b}_c\right), \\
\hat{\boldsymbol{y}}^{d_i}_t &= \text{Softmax}\left(\boldsymbol{W}_t e^{d_i} + \boldsymbol{b}_t\right),
\end{aligned} \tag{8}
$$

We set the threshold at 0.5 to determine the inclusion of the labels. The labels corresponding to the predicted values that exceed the threshold are denoted as $\hat{Y}^{d_i}_a$ and $\hat{Y}^{d_i}_c$, where $\hat{Y}^{d_i}_a \subset \boldsymbol{Y}_a$, $\hat{Y}^{d_i}_c \subset \boldsymbol{Y}_c$, $\hat{Y}^{d_i}_t = \arg\max \hat{\boldsymbol{y}}^{d_i}_t$, $\hat{Y}^{d_i}_t \in \boldsymbol{Y}_t$. $W_*$ and $b_*$ are the trainable parameters.

**Training.** To train this model, we use cross-entropy loss function for each subtask between the ground truth $y$ and prediction

probability of $\hat{y}$, and take the weighted sum as an overall loss by:

$$\mathcal{L} = -\lambda_a \sum_{j=1}^{|Y_a|} \boldsymbol{y}_{a,j} \log\left(\hat{\boldsymbol{y}}_{a,j}\right) - \lambda_c \sum_{j=1}^{|Y_c|} \boldsymbol{y}_{c,j} \log\left(\hat{\boldsymbol{y}}_{c,j}\right) - \lambda_t \sum_{j=1}^{|Y_t|} \boldsymbol{y}_{t,j} \log\left(\hat{\boldsymbol{y}}_{t,j}\right), \tag{9}$$

where $|Y_*|$ denotes the number of labels for each subtask, and $\lambda_*$ is the weight factor which is the hyperparameter for each subtask.

## 5 Experiments

**Datasets.** CAIL2024-DRDZ is the official dataset for the CAIL 2024 Multi-Defendant, Multi-Charge Judgment Prediction Competition[2]. It is derived from publicly available, high-quality court judgments in cases involving multiple defendants and includes the findings of fact, a list of all defendants involved in each case, and the corresponding sentencing outcomes. Table 1 presents the statistical information of the CAIL2024-DRDZ dataset.

**Experimental setting.** We select the law articles at article level for prediction. For the sake of comparison, we have used the sections of the baseline that mention the defendant's name, and have consistently employed BERT [1] as the base encoder. For training of PositionalLJP, the learning rate of the Adam optimizer is initialized as $10^{-5}$. We utilize PyTorch to implement the proposed model and train it on a server with H100 GPU, and train every model for 128 epochs with batch size 32. The weights $\lambda_*$ are set as 1.

**Evaluation metrics.** Performance evaluation results are reported in terms of Accuracy (Acc.), Macro-Precision (MP), Macro-Recall (MR) and Macro-F1 (F1) at case level [4].

**Baselines.** Firstly, we select the state-of-the-art models for the single defendant LJP task. (1) TopJudge [16], a multi-task learning framework; (2) NeurJudge [15], a circumstance framework. Secondly, we select some PLMs. (3) mT5 [13], a multilingual model is pre-trained by converting various language tasks into "text-to-text" tasks; (4) Lawformer [12], A pre-training model for chinese legal long documents understanding. Thirdly, we select the multi-defendant related methods in the legal field. (5) Official baseline of CAIL2024-DRDZ [3]; (6) MAMD [7], a method which models defendant name by Gaussian distribution;(7) HRN-Second-level [6], a prompt method to incorporate the name into the input question;(8) MUD [11]: a prompt-based approach to combine defendant names with case facts;(9) MAGLJP-Conviction-agent [14]: using prompt in LLMs. Finally, to further validate the performance of each component in our model, we design some variants without Positional Encoding and Prompt representation.

**Overall performance.** Table 2 illustrates the overall performance of our proposed method. The experimental results show that the proposed PositionalLJP significantly outperforms the other baselines. The ablation study verified the effectiveness of the Positional Encoding and Prompt Representation components.

**Impact of the defendant number.** We have categorised the number of defendants into intervals to illustrate the relationship between the number of defendants and prediction accuracy, as shown in Figure 3. The experimental results indicate that our proposed PositionalLJP outperforms the baseline method and mitigates the long-tail distribution to some extent.

[2]http://cail.cipsc.org.cn/task_summit?raceID=6&cail_tag=2024

[3]https://github.com/china-ai-law-challenge/CAIL2024/tree/main/drdz

## 6 Conclusion

In this paper, we proposed a multi-defendant legal judgment prediction framework called PositionalLJP to address the challenge of separating facts. Specifically, we constructed three prompts to represent the facts of different defendants in order to achieve the effect of splitting the facts in semantic level, and used positional encoding to represent defendant information more accurately. The experiment verified the effectiveness of the proposed method.

## References


[1] 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, volume 1 (long and short papers)*. 4171–4186.

[2] Leilei Gan, Kun Kuang, Yi Yang, and Fei Wu. 2021. Judgment prediction via injecting legal knowledge into neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 12866–12874.

[3] Jia He, Mukund Rungta, David Koleczek, Arshdeep Sekhon, Franklin X Wang, and Sadid Hasan. 2024. Does prompt formatting have any impact on llm performance? *arXiv preprint arXiv:2411.10541* (2024).

[4] Wanhong Huang, Yi Feng, Chuanyi Li, Honghan Wu, Jidong Ge, and Vincent Ng. 2024. Cmdl: A large-scale chinese multi-defendant legal judgment prediction dataset. In *Findings of the Association for Computational Linguistics ACL 2024*. 5895–5906.

[5] Bingfeng Luo, Yansong Feng, Jianbo Xu, Xiang Zhang, and Dongyan Zhao. 2017. Learning to predict charges for criminal cases with legal basis. *arXiv preprint arXiv:1707.09168* (2017).

[6] Yougang Lyu, Jitai Hao, Zihan Wang, Kai Zhao, Shen Gao, Pengjie Ren, Zhumin Chen, Fang Wang, and Zhaochun Ren. 2023. Multi-defendant legal judgment prediction via hierarchical reasoning. *arXiv preprint arXiv:2312.05762* (2023).

[7] Sicheng Pan, Tun Lu, Ning Gu, Huajuan Zhang, and Chunlin Xu. 2019. Charge prediction for multi-defendant cases with multi-scale attention. In *CCF Conference on Computer Supported Cooperative Work and Social Computing*. Springer, 766–777.

[8] Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2024. Representation learning with large language models for recommendation. In *Proceedings of the ACM web conference 2024*. 3464–3475.

[9] Suthir Sriram, Nivethitha Vijayaraj, G Rajiv Krishna, S Vijay Bhanu, Gaurav Choudhary, and Thangavel Murugan. 2026. A Reusable Prompting Framework for Applying Large Language Models to Legal Tasks. *IEEE Access* 14 (2026), 3108–3129.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[11] Xiao Wei, Qi Xu, Hang Yu, Qian Liu, and Erik Cambria. 2024. Through the MUD: A multi-defendant charge prediction benchmark with linked crime elements. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2864–2878.

[12] Chaojun Xiao, Xueyu Hu, Zhiyuan Liu, Cunchao Tu, and Maosong Sun. 2021. Lawformer: A pre-trained language model for chinese legal long documents. *AI Open* 2 (2021), 79–84.

[13] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: Human language technologies*. 483–498.

[14] Weikang Yuan, Kaisong Song, Zhuoren Jiang, Junjie Cao, Yujie Zhang, Chengyuan Liu, Jun Lin, Ji Zhang, Kun Kuang, and Xiaozhong Liu. 2026. A multi-agent framework with legal event logic graph for multi-defendant legal judgment prediction. *Information Processing & Management* 63, 1 (2026), 104319.

[15] Linan Yue, Qi Liu, Binbin Jin, Han Wu, Kai Zhang, Yanqing An, Mingyue Cheng, Biao Yin, and Dayong Wu. 2021. Neurjudge: A circumstance-aware neural framework for legal judgment prediction. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 973–982.

[16] Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Chaojun Xiao, Zhiyuan Liu, and Maosong Sun. 2018. Legal judgment prediction via topological learning. In *Proceedings of the 2018 conference on empirical methods in natural language processing*. 3540–3549.

[17] Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. How does NLP benefit legal system: A summary of legal artificial intelligence. *arXiv preprint arXiv:2004.12158* (2020).