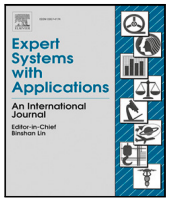
Expert Systems With Applications 238 (2024) 122103

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



Topology-aware Multi-task Learning Framework for Civil Case Judgment Prediction

Yuquan Le, Sheng Xiao *, Zheng Xiao, Kenli Li

College of Computer Science and Electronic Engineering, Hunan University, Changsha, China

ARTICLE INFO

Keywords:
Legal Artificial Intelligence
Civil case judgment prediction
Pre-trained language models
Multi-task learning framework
Topological dependencies

ABSTRACT

The civil case judgment prediction (CCJP) task involves automatically determining whether the plea of a plaintiff should be supported by analyzing the given civil case materials. However, most existing studies usually rely on inadequate legal essential elements (*e.g.*, fact descriptions and pleas), and are specifically designed for single-cause scenarios. Consequently, these methods struggle to generalize effectively to real courts, where civil cases involve more complicated legal elements and numerous causes. To resolve the above limitations, we present a novel **T**opology-aware **M**ulti-task **L**earning framework, called TML. Concretely, TML adopts the transformer-family pre-trained language models (PLMs) as the backbone to capture the fine-grained semantic interactions among various legal elements. To exploit the structural information of the case, we collocate distinct special tokens for each legal element, and then extract the features of the case from different perspectives. Furthermore, to address multiple-cause scenarios, TML incorporates a multi-task learning paradigm to simultaneously predict multiple civil judicial subtasks (*e.g.*, civil causes, civil law articles and final judgment of pleas). To utilize topological dependencies among subtasks, three parameter-free retrievers are integrated to establish inter-task connections. Extensive experiments are conducted on a real-world dataset, and the experimental results show the effectiveness of our proposed method.

1. Introduction

Due to the rapid progression of Artificial Intelligence technologies (*e.g.*, CNNs (Li, Liu, Yang, Peng, & Zhou, 2022), RNNs (Yu, Si, Hu, & Zhang, 2019), GNNs (Weng, Zhou, Li, Tan, & Li, 2022; Zhou et al., 2020) and Transformer (Vaswani et al., 2017)), it has promoted the development of many fields, such as transportation (Chen et al., 2020), industrial information (Liu, Li, Li, & Buyya, 2017; Zhong et al., 2021), biological information (Yang, Wang, Zhou, Wei, & Peng, 2022) and law (Feng, Li, & Ng, 2022). The goal of Legal Artificial Intelligence (LegalAI) (Cardellino, Teruel, Alemany, & Villata, 2017; Kanapala, Pal, & Pamula, 2019; Turtle, 1995; Yu et al., 2022) is to utilize AI technologies, especially natural language processing (Zhong et al., 2020), to assist legal tasks in real-world scenarios. The CCJP task plays a significant role in LegalAI. Automated judgment prediction of civil cases holds the notable potential to promote the productivity of judicial practitioners, making it a subject of substantial interest to both legal professionals and AI researchers in recent years. Given civil case materials, including fact descriptions and relevant legal essential elements (*e.g.*, litigant statements and law articles), CCJP aims to automatically determine whether the particular plea made by the plaintiff should be upheld or dismissed by the court.

Early works (Long, Tu, Liu, & Sun, 2019; Ma et al., 2021) primarily pay attention to exploiting the distributional representation of textual semantics through designing a series of neural networks, and have made notable progress. For example, Long et al. (2019) depict a legal reading comprehension paradigm that captures semantic interactions among fact descriptions, pleas, and law articles for divorce cases. Ma et al. (2021) develop the MSJudge model and utilize claims, court debate, and fact descriptions for private lending cases. Unfortunately, these methodologies encounter difficulties with generalization to actual court trials due to two primary challenges: **complicated legal elements of civil cases** and **numerous civil causes in real world**. On the one hand, some crucial legal elements are overlooked, such as litigant statements or law articles. The statements of the plaintiff and the defendant usually contain the case controversies (Zou, 2010), while law articles are the basis for the judicial of the civil cases (Long et al., 2019). Ignoring these vital legal elements hinders the model from making reliable predictions. On the other hand, they are specifically designed for single-cause situations. Real-world scenarios often involve numerous causes, such as over 900 types of civil causes in China[1]. Models specifically designed for a single type of cause may not be

* Corresponding author.
E-mail addresses: leyuquan@hnu.edu.cn (Y. Le), xiaosheng@hnu.edu.cn (S. Xiao), zxiao@hnu.edu.cn (Z. Xiao), lkl@hnu.edu.cn (K. Li).

[1] https://www.court.gov.cn/shenpan-xiangqing-282031.html.

https://doi.org/10.1016/j.eswa.2023.122103
Received 12 July 2023; Received in revised form 5 October 2023; Accepted 9 October 2023
Available online 11 October 2023

Fig. 1. A simplified running example from the CCJP dataset. This dataset consists of three subtasks: CCP, CLAP, and FJP. The CCP aims to predict cause by examining the litigant statements (e.g., the claim and the pleas from a plaintiff and the argument from a defendant). The CLAP aims to decide the violation of law articles based on the provided fact description. FJP aims to determine whether to support the specific plea based on the given litigant statements and the fact description. This case contains two pleas of the plaintiff.

sufficient for addressing complex situations where multiple causes are involved.

Recent research (Zhao et al., 2022) has made strides in addressing the above limitations regarding CCJP as the multi-task learning problem. As illustrated in Fig. 1, they formalize CCJP into three subtasks: Civil Cause Prediction (CCP), Civil Law Articles Prediction (CLAP), and Final Judgment Prediction (FJP). CCP aims to predict causes by examining the litigant statements (*e.g.*, the pleas and the claim from a plaintiff and the argument from a defendant). CLAP seeks to decide the violation of law articles based on the provided fact description. FJP aims to determine whether to support the pleas based on the given litigant statements and the fact description. In order to simultaneously handle multiple subtasks, they put forward a multi-task learning approach. They utilize co-attention mechanism (Xiong, Zhong, & Socher, 2016) to capture the interaction features of facts and claims, and facts and arguments separately, which may result in the insufficient understanding of complicated legal elements, leading to suboptimal results (*e.g.*, the Macro-F1 of CCP only reaches 58.9%, while FJP's Macro-F1 is only 45.8%). Therefore, the CCJP task is underexplored and needs further exploration.

Over the past years, transformer-based PLMs have swept through many fields (Devlin, Chang, Lee, & Toutanova, 2019) and have achieved impressive results. PLMs also shine in legal domain tasks (Chalkidis, Fergadiotis, Malakasiotis, Aletras, & Androutsopoulos, 2020; Limsopatham, 2021; Xiao, Hu, Liu, Tu, & Sun, 2021), such as legal named entity recognition (Păi, Mitrofan, Gasan, Coneschi, & Ianov, 2021) and legal textual entailment (Wehnert, Dureja, Kutty, Sudhi, & De Luca, 2022). However, there are few ways utilizing the powerful text representation capabilities of PLMs to handle CCJP task. We argue that there are at least two points to be considered for the effective mining of PTMs to handle CCJP task:

Modeling of complex legal elements. As illustrated in Fig. 1, civil cases typically include complex legal elements, such as claims, pleas, arguments, and fact descriptions. Specifically, the fact description implies the basic facts of the case. Combining details of claims and arguments makes a more lucid comprehension of case controversies possible. The plea comprises a detailed description of the plaintiff's pleadings, and the judge needs to decide whether to support it. Failure to exhaustively comprehend respective parts of the civil case structure is likely to impede accurate prognostications.

Establish the inter-task connections between subtasks. In practical scenarios, there exist topological dependencies between different subtasks, and effective utilization of these relationships can be beneficial to CCJP. On the one hand, the cause can narrow the prediction of the law articles due to its tendency to be related to some specific articles (Zhao, Yue et al., 2022). Fig. 2(a), (b) and (c) show the co-occurrence statistics among three different causes and law articles from a real-world dataset[2]. These figures show that these causes usually manifest a significant co-occurrence ratio with the top eight most frequently co-occurring articles, amounting to more than 80%. It is worth emphasizing that the dataset encompasses 300 law articles. On the other hand, law articles are the reference point for the decision (Zhao, Yue et al., 2022). As shown in Fig. 1, the civil case describes that the accused party borrows funds from the prosecution, and issues the promissory note with a repayment date, yet neglects to fulfill their obligation. The case involves the *Article 60*, which stipulates that the parties should fulfill their obligations, and the *Article 207*, which stipulates that the borrower needs to pay late interest if he fails to repay the loan according to the agreed period, as shown in Fig. 2(c). Combined with the information from the relevant law articles, we can infer that the plea (*"…repay the plaintiff's load of 6,000 yuan and pay interest…"*) should be supported.

Inspired by the aforementioned findings, this paper presents a novel **T**opology-aware **M**ulti-task **L**earning framework, called TML, for CCJP. It utilizes a more holistic set of legal elements and possesses the ability to deal with multiple-cause scenarios via a multi-task learning paradigm to predict multiple civil judicial subtasks simultaneously. Concretely, TML is composed of three primary modules, including the civil cause prediction module (CCPM), the civil law articles prediction module (CLAPM), and the final judgment prediction module (FJPM), intended for different subtasks. We adopt transformer-family pre-trained language model (*e.g.*, Lawformer (Xiao et al., 2021)[3]) as the backbone to capture the fine-grained semantic interactions among different legal elements, and employ siamese architecture (Chopra, Hadsell, & LeCun, 2005) to share encoder parameters across different modules. We collocate special tokens for legal elements to endow the ability of TML to perceive the structural information of the case, which can better extract the features of the case from different perspectives. Additionally, we devise three parameter-free retrievers to inject the topological dependencies among different subtasks. Cause retriever aims to explore the linguistic knowledge of corresponding cause labels to assist law articles prediction, while article retriever aims to utilize legal knowledge of applicable law articles to aid the final judgment of pleas. We carry out a series of experiments on a real-world dataset, and the experimental results demonstrate the effectiveness of TML. We summarize the contributions of this work as follows:

- We present a novel Topology-aware Multi-task Learning framework to address multiple-cause scenarios of CCJP better. To establish inter-task connections, we devise three parameter-free retrievers to explore topological dependencies among subtasks.

[2] https://github.com/LiliizZ/CPEE/blob/main/CPEE_dataset.zip.

[3] The Lawformer is Longformer-based (Beltagy, Peters, & Cohan, 2020) pre-trained language model, which is selected as backbone thanks to its support for long text encoding and has been pre-trained on legal texts. Without loss of generality, other Longformer-based backbones can also replace encoder and thus integrate seamlessly into our framework.

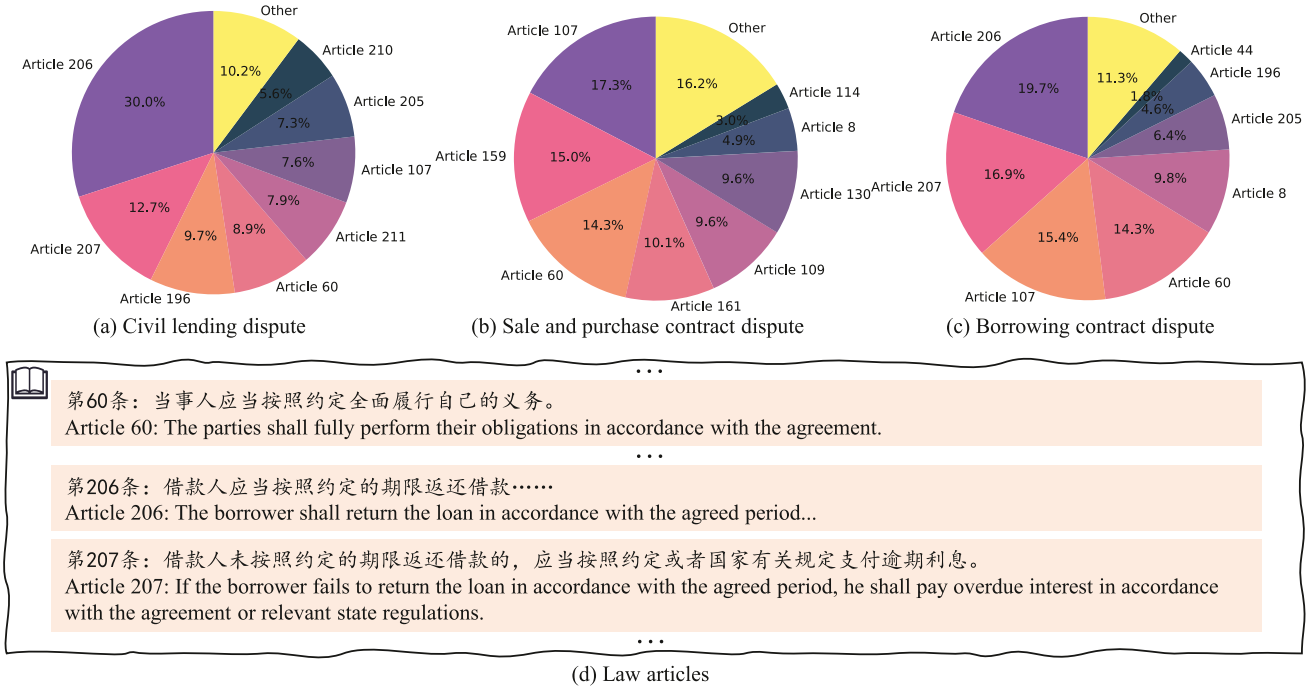


Fig. 2. The motivation figure of the proposed method. Figs. (a) (b) and (c) show the pie charts of high-frequency co-occurrence statistics for specific cases, respectively. Fig. (d) shows the text description of the law articles.

- We adopt transformer-based PTMs to capture fine-grained semantic interaction among complicated legal elements. To utilize the structural information of the case, we collocate distinct special tokens for each legal element, and extract the case features from different perspectives.
- Numerous experiments are carried out on a public real-world dataset. TML achieves competitive performance compared to the currently published state-of-the-art method. Qualitative and quantitative experiments show the effectiveness of utilizing cause and law article information. The low-resource setting experiment demonstrates that TML is more advantageous in that setting.

The remaining parts of this work are organized according to the following layout. In Section 2, we summarize the related works. In Section 3, we introduce some notations to formalize the CCJP problem. In Section 4, the overall framework of TML is presented in detail. In Section 5, we conduct adequate experiments to answer five research questions. In Section 6, we discuss ethical concerns of TML. In Section 7, we conclude this study as well as provide several potential directions for future research.

2. Related work

Legal Judgment Prediction. Legal Judgment Prediction (LJP) is the most influential task in LegalAI, widespread attention from researchers around the world (Ashley, 2019; Branting et al., 2021; Feng et al., 2022; Katz, Bommarito, & Blackman, 2017; Medvedeva, 2022; Medvedeva, Wieling, & Vols, 2023). In countries with the civil law system, existing LJP methodologies primarily focus on criminal case judgment prediction (Feng et al., 2022) and civil case judgment prediction (Long et al., 2019; Ma et al., 2021; Zhao, Yue et al., 2022). Criminal case judgment prediction aims to determine relevant criminal judgment outcomes (*e.g.*, charges, articles, terms of penalty) based on provided case materials (*e.g.*, fact descriptions). A series of deep learning based methods have been presented to predict the charges (Hu, Li, Tu, Liu, & Sun, 2018; Le et al., 2020, 2022; Zhao, Guan, Xu, Zhao and Chen, 2022), articles (Zhang, Wang, Tan, & Li, 2019), and terms of penalty (Bi, Zhou, Pan, & Qi, 2022; Chen, Cai, Dai, Dai, & Ding, 2019) separately or to predict multiple criminal verdict outcomes (Feng et al., 2022) simultaneously. Literatures (Long et al., 2019; Zhou, Zhang, Liu, Sun, & Si, 2019) on civil case judgment prediction are more relevant to our work. The existing works focus on exploring the characteristics and structure of civil cases. For example, Long et al. (2019) utilize multiple inputs, which include fact descriptions, pleas, and law articles, for divorce dispute via a reading comprehension model. Zhou et al. (2019) explore multiview contexts (*e.g.*, past behavior information of seller and buyer) for e-commerce transactions dispute. Gan, Kuang, Yang, and Wu (2021) deal with private lending dispute cases by introducing declarative legal knowledge as the first-order logic rules. Ma et al. (2021) propose a multi-stage judgment predictor to model claims, court debate, and fact descriptions for private lending dispute. These works only focus on dealing with single-cause scenarios, and may fail to deal with real-world scenarios involving multiple-cause. Zhao et al. (2021) introduce external explanations knowledge of each cause to assist judgment of various causes. However, they ignore the judicial logic of the real judicial scenarios. The work that is most relevant to our method is the literature (Zhao, Yue et al., 2022), since we both consider the topological among different subtasks. Nevertheless, unlike that they utilize co-attention to capture interaction between features of fact descriptions and claims, and between features of fact descriptions and arguments separately, we adopt transformer-family PLMs to capture the fine-grained feature interaction between different legal elements.

Transformer. The transformer has been widely researched in natural language processing community (Wolf et al., 2020), owing to its powerful capability of capturing the relationship between arbitrary token pairs in a sequence using self-attention mechanism (Vaswani et al., 2017). In recent years, significant progress has been achieved in PLMs using large-scale unsupervised data as a corpus, with the transformer as the fundamental building block (Qiu et al., 2020; Zhao et al., 2023). One of the most popular PLMs is BERT (Devlin et al., 2019), and it has been trained on data from a wide range of domains, including the legal domain (Chalkidis et al., 2020). A major drawback of BERT is its limited input length of 512 tokens. To overcome this limitation, the Longformer (Beltagy et al., 2020) combines local sliding window

Table 1
The main mathematical symbols in this paper.

Notation	Description
$\boldsymbol{T}^c = \{w_1^c, \ldots, w_{\ell_c}^c\}$	The textual description of claims of plaintiff
$\mathcal{T}^p = \{\boldsymbol{T}^{p_1}, \ldots, \boldsymbol{T}^{p_k}\}$	The collection of the k pleas of the plaintiff
$\boldsymbol{T}^{p_i} = \{w_1^{p_i}, \ldots, w_{\ell_p}^{p_i}\}$	The textual description of i-*th* pleas of the plaintiff
$\boldsymbol{T}^a = \{w_1^a, \ldots, w_{\ell_a}^a\}$	The textual description of the defendant's arguments
$\boldsymbol{T}^f = \{w_1^f, \ldots, w_{\ell_f}^f\}$	The textual description of fact descriptions
$\mathcal{T}^\ell = \{\boldsymbol{T}^c, \mathcal{T}^p, \boldsymbol{T}^a\}$	The collection of litigant statements
$\mathcal{T}^{ga} = \{\boldsymbol{T}^{ga_1}, \ldots, \boldsymbol{T}^{ga_{n_g}}\}$	The collection of all n_g general law articles
$\boldsymbol{T}^{ga_i} = \{w_1^{ga_i}, \ldots, w_{\ell_{ga}}^{ga_i}\}$	The textual description of i-*th* general law article
$\mathcal{T}^{sa} = \{\boldsymbol{T}^{sa_1}, \ldots, \boldsymbol{T}^{sa_{n_s}}\}$	The collection of all n_s general law articles
$\boldsymbol{T}^{sa_i} = \{w_1^{sa_i}, \ldots, w_{\ell_{sa}}^{sa_i}\}$	The textual description of i-*th* specific law article
$\mathcal{T}^{c\ell} = \{\boldsymbol{T}^{c\ell_1}, \ldots, \boldsymbol{T}^{c\ell_{n_c}}\}$	The collection of all n_c cause labels
$\boldsymbol{T}^{c\ell_i} = \{w_1^{c\ell_i}, \ldots, w_{\ell_{c\ell}}^{c\ell_i}\}$	The textual description of i-*th* cause label
$\mathbf{Y}_c = \{\mathbf{y}_c^1, \ldots, \mathbf{y}_c^{n_c}\}$	The category coding of the cause labels
$\mathbf{Y}_{ga} = \{\mathbf{y}_{ga}^1, \ldots, \mathbf{y}_{ga}^{n_g}\}$	The category coding of the general article labels
$\mathbf{Y}_{sa} = \{\mathbf{y}_{sa}^1, \ldots, \mathbf{y}_{sa}^{n_s}\}$	The category coding of the specific article labels
$\mathbf{Y}_p = \{\mathbf{y}_p^1, \ldots, \mathbf{y}_p^{n_p}\}$	The category coding of the plea labels

attention with global task-motivated full attention, instead of relying on the original self-attention mechanism. The Longformer model can support thousands of tokens as input. The Lawformer (Xiao et al., 2021) is a Longformer-based PLMs pre-trained on legal text. Thanks to its long legal textual encoding support, this paper adopts Lawformer as the backbone. To enable Lawformer to perceive the structural information of civil cases, we collocate special tokens for legal elements, and thus can better extract the case features from different perspectives.

3. Preliminary

To facilitate the understanding of the TML framework, this section introduces some notations to formalize the CCJP task. Following the previous work (Zhao, Yue et al., 2022), the CLAP is decomposed into civil general law article prediction (CGLAP) and civil specific law article prediction (CSLAP). Formalizally, given the litigant statements $\mathcal{T}^\ell$, including claim $\boldsymbol{T}^c$, pleas $\mathcal{T}^p$ and argument $\boldsymbol{T}^a$, the CCP aims to predict the cause $\hat{\mathbf{y}}_c$ of case. Given the fact description $\boldsymbol{T}^f$, the CGLAP and CSLAP aim to determine corresponding general articles $\hat{\mathbf{y}}_{ga}$ and special articles $\hat{\mathbf{y}}_{sa}$. Given the litigant statements and fact description, the FJP aims to determine whether pleas made by the plaintiff should be upheld or dismissed $\hat{\mathbf{y}}_p$. Formulaically, CCJP aims to find the most outcomes of these subtasks (*e.g.*, $\hat{\mathbf{y}}_c$, $\hat{\mathbf{y}}_{ga}$, $\hat{\mathbf{y}}_{sa}$ and $\hat{\mathbf{y}}_p$), by maximizing the probability over the candidate spaces (*e.g.*, $\mathbf{Y}_c$, $\mathbf{Y}_{ga}$, $\mathbf{Y}_{sa}$ and $\mathbf{Y}_p$) as Eq. (1).

$$\{\hat{\mathbf{y}}_c, \hat{\mathbf{y}}_{ga}, \hat{\mathbf{y}}_{sa}, \hat{\mathbf{y}}_p\} \leftarrow \underset{\substack{\mathbf{y}_c \in \mathbf{Y}_c, \mathbf{y}_{ga} \in \mathbf{Y}_{ga} \\ \mathbf{y}_{sa} \in \mathbf{Y}_{sa}, \mathbf{y}_p \in \mathbf{Y}_p}}{\arg\max} P(\mathbf{y}_c, \mathbf{y}_{ga}, \mathbf{y}_{sa}, \mathbf{y}_p | \mathcal{T}^\ell, \boldsymbol{T}^f). \tag{1}$$

Explicitly, the CCP and FJP are formalized as the multi-class text classification problem, while CGLAP and CSLAP are formalized as the multi-label text classification problem. To utilize topological dependencies among subtasks, we present the TML framework as shown in Fig. 3. TML handles civil judicial probability $P(\mathbf{y}_c, \mathbf{y}_{ga}, \mathbf{y}_{sa}, \mathbf{y}_p | \mathcal{T}^\ell, \boldsymbol{T}^f)$ using Eq. (2).

$$P(\mathbf{y}_p | \mathcal{T}^\ell, \boldsymbol{T}^f, \mathcal{T}^{ga}, \mathcal{T}^{sa}, \hat{\mathbf{y}}_{ga}, \hat{\mathbf{y}}_{sa}) P(\mathbf{y}_{ga}, \mathbf{y}_{sa} | \boldsymbol{T}^f, \mathcal{T}^{c\ell}, \hat{\mathbf{y}}_c) P(\mathbf{y}_c | \mathcal{T}^\ell). \tag{2}$$

In the TML framework, the linguistic knowledge of corresponding causes (*e.g.*, $\mathcal{T}^{c\ell}$ and $\hat{\mathbf{y}}_c$) is utilized to aid law articles prediction (*e.g.*, $\hat{\mathbf{y}}_{ga}$ and $\hat{\mathbf{y}}_{sa}$). The legal knowledge of applicable law articles (*e.g.*, $\mathcal{T}^{ga}$, $\mathcal{T}^{sa}$, $\hat{\mathbf{y}}_{ga}$ and $\hat{\mathbf{y}}_{sa}$) is explored to help final judgment prediction of pleas (*e.g.*, $\hat{\mathbf{y}}_p$). Considering that in practical scenarios, civil cases often contain one to multiple pleas, LML predicts FJP (*e.g.*, $\hat{\mathbf{y}}_{p_i} \in \mathbf{Y}_p$) of each plea (*e.g.*, $\boldsymbol{T}^{p_i}$, $i \in [1, k]$) respectively. Please note that k varies in different cases. Detailed symbol descriptions can be found in Table 1.

4. Model

This section provides a detailed introduction to the proposed TML framework, as shown in Fig. 3. TML is tailored for CCJP and primarily includes three modules to handle different subtasks separately. Concretely, CCPM concatenates the claim, plea, and argument as input to predict causes. CLAPM utilizes cause retriever to obtain the linguistic knowledge of high-likelihood cause labels, and concatenates it with the fact description as input to predict law articles. FJPM adopts article retriever to obtain the high-likelihood law articles, and splices them in conjunction with the other relevant legal elements to predict the final judgment of each plea.

4.1. Civil cause prediction module

The CCPM composes of the CCP encoder and the CCP prediction layer. The CCP encoder is designed to encode relevant legal elements of the CCP task, while the CCP prediction layer aims to decide the cause by leveraging the encoded features.

Instead of focusing on the isolated connections between facts and claims, and facts and arguments (Zhao, Yue et al., 2022), we apply transformer-family PLMs (*e.g.*, Lawformer) as CCP encoder to capture the fine-grained features interaction among the three using its built-in self-attention mechanism variant (Vaswani et al., 2017). Formalizally, given the token sequence of claim $\boldsymbol{T}^c$, argument $\boldsymbol{T}^a$ and one or multiple pleas $\mathcal{T}^p$ of a specific civil case, we concatenate legal elements according to the style of civil case[4] as input to the CCPM, as follows:

$$\begin{aligned} \boldsymbol{T}_{ccp} = & [\text{CLS}][\text{CLAIM}] w_1^c, \ldots, w_{\ell_c - 2}^c \\ & [\text{PLEA}] w_1^{p_i}, w_2^{p_i}, \ldots, w_{\ell_p * k - 1}^{p_i} \\ & [\text{ARGU}] w_1^a, w_2^a, \ldots, w_{\ell_a - 2}^a [\text{SEP}], \end{aligned} \tag{3}$$

where ℓ_c, ℓ_p and ℓ_a are the maximum token sequence length of the claim, pleas, and argument, respectively. The k denotes the number of pleas of a specific case. The special token [CLS] represents the commencement of the entire text, and [SEP] represents the termination of each type of legal element. The special tokens are added different legal elements to perceive the case structure. For example, the special token [CLAIM] represents the beginning of a claim, as detailed in Section 5.1.4. We add the $\boldsymbol{T}_{ccp}$'s token representation $\mathbf{V}_{ccp}$ and its position encoding $\mathbf{PE}_{ccp}$ to obtain the input of CCPM, which is as follows:

$$\begin{aligned} \mathbf{E}_{ccp} &= \mathbf{V}_{ccp} + \mathbf{PE}_{ccp}, \\ \mathbf{H}_{ccp} &= \text{Encoder}_{ccp}(\mathbf{E}_{ccp}), \end{aligned} \tag{4}$$

where $\mathbf{H}_{ccp}$ is the token features after encoding. The encoder is able to capture the fine-grained semantic interaction between claims, pleas, and arguments. Information from different legal element perspectives benefits for civil judgment tasks (Zhao, Yue et al., 2022). Instead of using the representation of [CLS] as the overall case feature, we first decompose $\mathbf{H}_{ccp}$ into representations of different legal elements, including $\mathbf{H}_{ccp}^c$, $\mathbf{H}_{ccp}^p$, and $\mathbf{H}_{ccp}^a$. Then, the pool operation is used to obtain the representations of different legal element perspectives, such as the features of claims is obtained through $\mathbf{U}_{ccp}^c = \text{Pool}(\mathbf{H}_{ccp}^c)$. Finally, the CCP prediction layer aims to predict the cause based on the representations of different perspectives, with the following equation:

$$\hat{\mathbf{y}}_c = \sigma(\text{MLP}_{ccp}(\mathbf{U}_{ccp}^c \parallel \mathbf{U}_{ccp}^p \parallel \mathbf{U}_{ccp}^a)), \tag{5}$$

where $\text{Pool}(\cdot)$ is the pooling operation, and this work uses average pooling for simplicity. The $\parallel$ represents the concatenation operation, and $\sigma(\cdot)$ denotes the softmax function.

[4] https://www.court.gov.cn/susong.html.

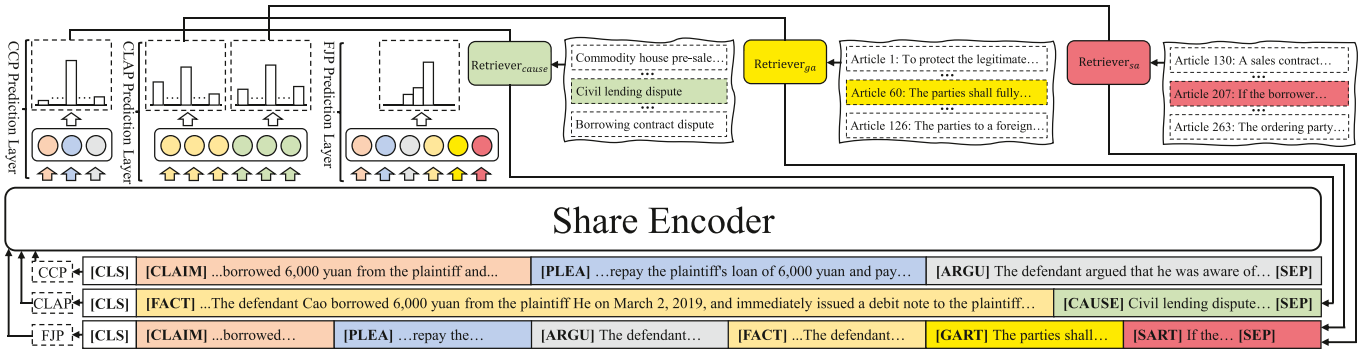


Fig. 3. The overview of the TML framework. TML consists of three modules (CCPM, CLAPM, and FJPM) to deal with different subtasks separately. Concretely, given the claim, the pleas, and the argument of a civil case, the CCPM aims to determine cause. The CLAPM utilizes cause retriever to obtain the linguistic knowledge of high-likelihood cause labels, and concatenates it with the fact description as input to predict law articles. FJPM adopts article retriever to obtain the high-likelihood law articles, and splices them in conjunction with the other relevant legal elements (e.g., claims, pleas, arguments, and fact descriptions) to determine the final judgment of each plea.

4.2. Civil law articles prediction module

The CLAPM contains the cause retriever, the CLAP encoder, and the CLAP prediction layer. The cause retriever aims to retrieve the linguistic knowledge of high-likelihood cause labels. The CLAP encoder is designed to encode the fact description and candidate causes, and the CLAP prediction layer aims to predict the law articles by leveraging the encoded features.

As mentioned before, there exist relationships between causes and law articles. Instead of limiting the scope of the law articles by the cause (Zhao, Yue et al., 2022), we establish a connection with the law articles through the linguistic semantics of the cause textual description. To this end, we devise a cause retriever to retrieve candidate causes as follows:

$$\mathcal{T}^{c\ell}_{topk} = \text{Retriever}_{cause}(\hat{\mathbf{y}}_c, \mathcal{T}^{c\ell}), \tag{6}$$

where $\mathcal{T}^{c\ell}_{topk} = \{\boldsymbol{T}^{c\ell_i}\}$, $i \in \text{Top}_k(\hat{\mathbf{y}}_c, k_{c\ell})$. The $\text{Top}_k(\hat{\mathbf{y}}_c, k_{c\ell})$ function aims to return the position corresponding to the largest top $k_{c\ell}$ values in $\hat{\mathbf{y}}_c$. The $\text{Retriever}_{cause}(\cdot)$ is a parameter-free retrieval layer.

The CLAP encoder adopts the fact description and the linguistic knowledge of candidate causes as the input as following:

$$\begin{aligned}\boldsymbol{T}_{clap} =& [\text{CLS}][\text{FACT}]w^f_1, \ldots, w^f_{\ell_f-2} \\ & [\text{CAUSE}]w^{c\ell_i}_1, \ldots, w^{c\ell_i}_{\ell_{c\ell}*k_{c\ell}-2}[\text{SEP}],\end{aligned} \tag{7}$$

where [FACT] special token indicates the beginning of the fact description and [CAUSE] indicates the cause textual description. We add $\boldsymbol{T}_{clap}$'s token representation $\mathbf{V}_{clap}$ and its position encoding $\mathbf{PE}_{clap}$ as input of CLAPM, which is as follows:

$$\begin{aligned}\mathbf{E}_{clap} &= \mathbf{V}_{clap} + \mathbf{PE}_{clap}, \\ \mathbf{H}_{clap} &= \text{Encoder}_{clap}(\mathbf{E}_{clap}),\end{aligned} \tag{8}$$

where $\mathbf{H}_{clap}$ is the token feature after encoding. Similar to CCP task, we slice $\mathbf{H}_{clap}$ into representations of different perspectives, including $\mathbf{H}^f_{clap}$, $\mathbf{H}^{c\ell}_{clap}$. Then, we use pool operation to obtain representations of different perspectives (*e.g*, the feature of the fact description is gained by $\mathbf{U}^f_{clap} = \text{Pool}(\mathbf{H}^f_{clap})$). Finally, the CLAP prediction layer predicts the law articles through the representations of different perspectives, with the following equations:

$$\begin{aligned}\hat{\mathbf{y}}_{ga} &= \sigma(\text{MLP}_{cglap}(\mathbf{U}^{c\ell}_{clap} \parallel \mathbf{U}^f_{clap})), \\ \hat{\mathbf{y}}_{sa} &= \sigma(\text{MLP}_{cslap}(\mathbf{U}^{c\ell}_{clap} \parallel \mathbf{U}^f_{clap})).\end{aligned} \tag{9}$$

4.3. Final judgment prediction module

The FJPM contains the article retriever, the FJP encoder, and the FJP prediction layer. The article retriever aims to retrieve high-likelihood articles relevant to the specific case. The formula is as follows:

$$\begin{aligned}\mathcal{T}^{ga}_{topk} &= \text{Retriever}_{ga}(\hat{\mathbf{y}}_{ga}, \mathcal{T}^{ga}), \\ \mathcal{T}^{sa}_{topk} &= \text{Retriever}_{sa}(\hat{\mathbf{y}}_{sa}, \mathcal{T}^{sa}),\end{aligned} \tag{10}$$

where $\mathcal{T}^{ga}_{topk} = \{\boldsymbol{T}^{ga_i}\}$, $i \in \text{Top}_k(\hat{\mathbf{y}}_{ga}, k_{ga})$, and $\mathcal{T}^{sa}_{topk} = \{\boldsymbol{T}^{sa_i}\}$, $i \in \text{Top}_k(\hat{\mathbf{y}}_{sa}, k_{sa})$. The $\text{Retriever}_{ga}(\cdot)$ is general law articles retriever, and the $\text{Retriever}_{sa}(\cdot)$ is special law articles retriever. The FJP encoder is designed to encode relevant input information of FJP. The formula is as follow:

$$\begin{aligned}\boldsymbol{T}_{fjp} =& [\text{CLS}][\text{CLAIM}]w^c_1, \ldots, w^c_{\ell_c-2}, \\ & [\text{PLEA}]w^{p_i}_1, w^{p_i}_2, \ldots, w^{p_i}_{\ell_p-1}, \\ & [\text{ARGU}]w^a_1, w^a_2, \ldots, w^a_{\ell_a-1}, \\ & [\text{FACT}]w^f_1, w^f_2, \ldots, w^f_{\ell_f-1}, \\ & [\text{GART}]w^{ga_i}_1, \ldots, w^{ga_i}_{\ell_{ga}*k_{ga}-1}, \\ & [\text{SART}]w^{sa_i}_1, \ldots, w^{sa_i}_{\ell_{sa}*k_{sa}-2}[\text{SEP}],\end{aligned} \tag{11}$$

where [GART] special token indicates the beginning of the general law article and [SART] indicates the beginning of the special law article. We add $\boldsymbol{T}_{fjp}$'s token representation $\mathbf{V}_{fjp}$ and its position encoding $\mathbf{PE}_{fjp}$ as input of CCPM, which is as follows:

$$\begin{aligned}\mathbf{E}_{fjp} &= \mathbf{V}_{fjp} + \mathbf{PE}_{fjp}, \\ \mathbf{H}_{fjp} &= \text{Encoder}_{fjp}(\mathbf{E}_{fjp}),\end{aligned} \tag{12}$$

where $\mathbf{H}_{fjp}$ is the token feature after encoding. Similar to CLAP task, we slice $\mathbf{H}_{fjp}$ into representations of different perspectives, $\mathbf{H}^c_{fjp}$, $\mathbf{H}^{p_i}_{fjp}$, $\mathbf{H}^a_{fjp}$, $\mathbf{H}^f_{fjp}$, $\mathbf{H}^{ga}_{fjp}$, $\mathbf{H}^{sa}_{fjp}$, respectively. Then we use pool operation to obtain representations of different perspectives, such as the feature of claims is obtained by $\mathbf{U}^c_{fjp} = \text{Pool}(\mathbf{H}^c_{fjp})$. Finally, the FJP prediction layer predicts the final judgment of each plea based on the representations of different perspectives, with the following equations:

$$\hat{y}_{p_i} = \sigma(\text{MLP}_{fjp}(\mathbf{U}^c_{fjp} \parallel \mathbf{U}^{p_i}_{fjp} \parallel \mathbf{U}^a_{fjp} \parallel \mathbf{U}^f_{fjp} \parallel \mathbf{U}^{ga}_{fjp} \parallel \mathbf{U}^{sa}_{fjp})). \tag{13}$$

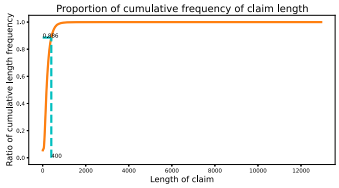
Civil cases often contain multiple pleas, and the number of pleas varies among different cases. For a case containing k pleas, FJPM executes k times, focusing on understanding one plea each time. We adopt a twin approach (Chopra et al., 2005) to design the FJP, the CLAP, and the CCP encoders, which share parameters but have different inputs.

4.4. Model training

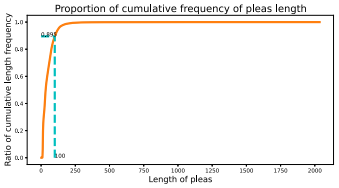
Our framework involves three different subtasks: the CCP, the CLAP, and the FJP. It mainly includes three parts of the loss function. The total

Table 2
The detailed statistical of dataset. #Case represents the number of civil cases, and #Pleas indicates the number of pleas of plaintiff.

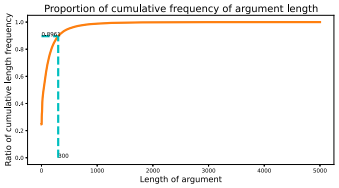
Datasets	#Cases	#Pleas	Causes	General articles	Specific articles	FJP	Average general articles	Average specific articles	Average pleas
Train	126,853	256,055	10	122	76	3	1.77	1.21	2.02
Valid	15,878	31,924	10	122	76	3	1.76	1.22	2.01
Test	15,894	32,024	10	122	76	3	1.75	1.22	2.02



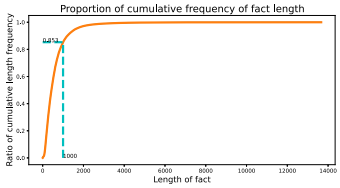
(a) The proportion of accumulation frequency of claims length distribution.



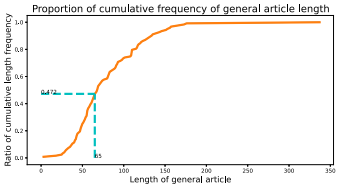
(b) The proportion of accumulation frequency of pleas length distribution.



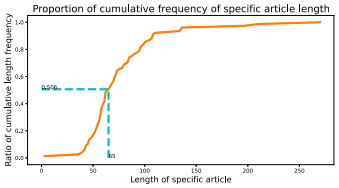
(c) The proportion of accumulation frequency of arguments length distribution.



(d) The proportion of accumulation frequency of fact descriptions length distribution.



(e) The proportion of accumulation frequency of general articles length distribution.



(f) The proportion of accumulation frequency of specific articles length distribution.

Fig. 4. The proportion accumulation frequency of the length distribution of different legal elements.

loss function is as follows:

$$\mathcal{L} = \alpha * \mathcal{L}_{ccp} + \beta * \mathcal{L}_{clap} + \gamma * \mathcal{L}_{fjp}, \tag{14}$$

where α, β, and γ are parameters to adjust the weight of loss function of different subtasks. The $\mathcal{L}_{ccp}$, $\mathcal{L}_{clap}$ and $\mathcal{L}_{fjp}$ are the loss function of the three subtasks, and the formulas are as follows:

$$\begin{aligned} \mathcal{L}_{ccp} &= \mathrm{CE}(\mathbf{y}_c, \hat{\mathbf{y}}_c), \\ \mathcal{L}_{clap} &= \mathrm{BCE}(\mathbf{y}_{ga}, \hat{\mathbf{y}}_{ga}) + \mathrm{BCE}(\mathbf{y}_{sa}, \hat{\mathbf{y}}_{sa}), \\ \mathcal{L}_{fjp} &= \mathrm{CE}(\mathbf{y}_{p_i}, \hat{\mathbf{y}}_{p_i}), \end{aligned} \tag{15}$$

where $\mathbf{y}_c$, $\mathbf{y}_{ga}$, $\mathbf{y}_{sa}$ and $\mathbf{y}_{p_i}$ are the ground truth of causes, general law articles, special law articles, and pleas, respectively. The $\hat{\mathbf{y}}_c$, $\hat{\mathbf{y}}_{ga}$, $\hat{\mathbf{y}}_{sa}$ and $\hat{\mathbf{y}}_{p_i}$ are the prediction label of causes, general law articles, special law articles, and pleas, respectively. The $\mathrm{BCE}(\cdot)$ denotes binary cross entropy loss, and the $\mathrm{CE}(\cdot)$ represents cross entropy loss.

5. Experiments

This section carries out numerous experiments, which aim to investigate the following five research questions:

- **RQ1**: How does the performance of our method in contrast to publicly published state-of-the-art methods?
- **RQ2**: What is the impact of the different modules of TML on performance?
- **RQ3**: Can the cause information be beneficial for law article prediction?
- **RQ4**: Can the law article help the final judgment prediction of pleas?
- **RQ5**: How does the TML perform in low-resource scenarios?

5.1. Experiment settings

5.1.1. Datasets

Similar to prior literature (Zhao, Yue et al., 2022), the real-world dataset[2] is selected as the benchmark to verify the effectiveness of TML. This dataset is collected from the China Judgments Online[5], and includes 158,625 civil cases. Each civil case is composed of case materials and multiple judgment results. The case materials comprise litigant statements (*e.g.*, claims, arguments, and pleas) and fact descriptions. The judgment results include cause, law articles, and final judgment of each plea[6]. According to the scope of application, the law articles are divided into special articles and general articles. Table 2 shows the detailed dataset statistics.

5.1.2. Evaluation metrics

Following the existing literature (Zhao, Yue et al., 2022), the Accuracy, Macro Precision (Mac-P), Macro Recell (Mac-R), and Macro F1 (Mac-F1) are adopted to measure the performance of TML in terms of CCP and FJP subtasks. The Micro Precision (Mic-P), Micro Recell (Mic-R), and Micro F1 (Mic-F1) are utilized to evaluate the performance of TML in terms of the CLAP subtask. Please note that Mac-F1 is more suitable for assessing the effectiveness of the model in imbalanced scenarios compared to Accuracy, as it takes the macro average of all classes.

5.1.3. Baselines

To evaluate the performance of the TML framework, we select many representative baselines, which can be classified into the following three categories.

- **Traditional method**, including Word2Vec+SVM, which adopts word2vec (Mikolov, Sutskever, Chen, Corrado, & Dean, 2013) to encode text and employs SVM (Cortes & Vapnik, 1995) as classifier.

[5] https://wenshu.court.gov.cn/.

[6] The raw cases present a semi-structured organization, and contain special typographical signals. The CPEE dataset construction process uses regular expressions to extract the relevant parts.

Table 3
The hyperparameters of TML in detail.

Hyper-parameter	Values
Epochs	15
Batch size	4
Accumulation steps	64
Optimizer	AdamW
Cost function	CE,BCE
Learning rate	5e−5
Sequence length of claims	400
Sequence length of arguments	300
Sequence length of each plea	100
Sequence length of fact descriptions	1000
Sequence length of each cause textual description	10
Sequence length of each general article	64
Sequence length of each specific article	64
Top-k high likelihood prediction causes	2
Top-k high likelihood prediction general articles	2
Top-k high likelihood prediction specific articles	2
Loss weights of CCP, CLAP, FJP	{1,1,3}

- **Pre-trained language models**, including BERT-Civil (Devlin et al., 2019) and Lawformer (Xiao et al., 2021). The BERT-Civil utilizes the transformer as the block and pre-train on large-scale civil cases data. The Lawformer is a popular Legal PLM and adopts the Longformer to break through the BERT input 512 length limit.
- **Approaches tailed for CCJP**, including AutoJudge (Long et al., 2019), MSJudge (Ma et al., 2021), CCJudge (Zhao et al., 2021) and CPEE (Zhao, Yue et al., 2022). The AutoJudge handles the final judgment of pleas via the reading comprehension paradigm. The MSJudge aims to deal with the final judgment of private lending debate dispute, and is adapted for CCJP by replacing debate data with claims and arguments. The CCJudge utilizes multiple information from different perspectives, and knowledge bases for FJP. The CPEE utilizes the logic of judicial and adopts comprehensive legal elements for CCJP. Note that this group's approaches can only handle FJP subtask except for the CPEE method, which can handle all subtasks simultaneously.

Similar to literature (Zhao, Yue et al., 2022), to evaluate performance on CCP subtask, we select some baselines as follows:

- **HMC** builds the decision tree to capture the hierarchy dependencies relationship among labels.
- **Hdltex** is the hierarchical classification approach, which adopts multiple deep learning models to generate hierarchy.
- **HARNN** is a hierarchical attention-based model, which aims to capture the relationship among hierarchical structure and texts.
- **Flat-cause** utilizes Bi-directional LSTM to encode texts and average pooling strategy to aggregate features.
- **Hie+fact** adopts fact descriptions as input instead of litigant statements.

5.1.4. Implemention details

We have leveraged PyTorch[7] deep learning framework to instantiate TML framework, with the legal PLM Lawformer[8] serving as our chosen encoder. The TML is trained on the designated training set, with the best-performing model being chosen based on the performance on the validation set. We adapt AdamW optimizer (Loshchilov & Hutter, 2019) with a learning rate of 5e–5. We adopt four 3090 NVIDIA GPUs for distributed training and utilize gradient accumulation techniques. We do not perform parameter space search experiments on the loss weight hyperparameters, due to the larger computational effort. During the experiments, we observe that FJP is more difficult to converge. Therefore the loss weights for FJP is set to 3, while the weights for CCP and CLAP are set to 1. The distribution of the sequence lengths of the different legal elements in the training set is shown in Fig. 4. The special tokens (*e.g.*, [CLAIM], [PLEA], [ARGU], [FACT], [GART], [SART] and [CAUSE]) are collocated to different type contents (*e.g.*, claims, pleas, arguments, fact descriptions, general articles, specific articles, and causes). The detailed parameter settings are shown in Table 3.

The setting of baselines is as follows. We have implemented Lawformer with the goal of jointly predicting multiple subtasks of the CCJP. We concatenate the disparate legal elements of the original text for input, and select their collective representation at [CLS] for prediction, while maintaining other hyperparameters consistent with TML to ensure fairness. The performance of other baselines is extracted from the literature (Zhao, Yue et al., 2022).

5.2. Overall performance comparison (RQ1)

This section investigates how TML performs compared to competitive baselines. Table 4 shows the main experimental results on CCP and FJP. Fig. 5 illustrates the main comparison results on CLAP.

5.2.1. Comparison in terms of FJP

We can draw the following conclusions from the left half of Table 4.

- For most baselines, accuracy is much higher than macro-F1, and macro-F1 performance is very poor. The possible reason for this phenomenon: the ratio of categories (*e.g.*, *supported pleas*, *partially supported pleas* and *rejected pleas*) of FJP is 5.8:1:1.7, which is an unbalanced classification task. These methods tend to make predictions that favor the high-frequency classes, thus revealing high accuracy and low macro-F1. An ideal model should guarantee the performance of both accuracy and macro-F1.
- CPEE achieves the best performance in the oriented-tasks design of the baselines. This indicates that the judicial logical relationship utilized by CPEE is beneficial.
- The Lawformer achieves the best Macro-F1 performance, while CPEE obtains the best accuracy performance in all baselines. Specifically, CPEE is 6.2% higher than Lawformer in terms of accuracy, but 19% lower than Lawformer in terms of macro-F1. Combining the two metrics, Lawformer performs better than CPEE, while CPEE tends to make predictions that favor the high-frequency category heavily. This experimental phenomenon illustrates that the powerful legal text comprehension capabilities of PLMs in the legal domain can effectively promote judgment prediction of pleas.
- To our surprise, our method is 2.2% lower in accuracy compared to CPEE. However, our method exceeds CPEE by 27.1% in terms of macro-F1. Meanwhile, TML surpasses Lawformer by 4% and 8.1% in terms of accuracy and macro-F1, respectively. This evidence shows the superiority of TML. One possible reason is that the corresponding law articles are beneficial for predicting the judgment of pleas.

5.2.2. Comparison in terms of CCP

The following observations can be found from the right of Table 4.

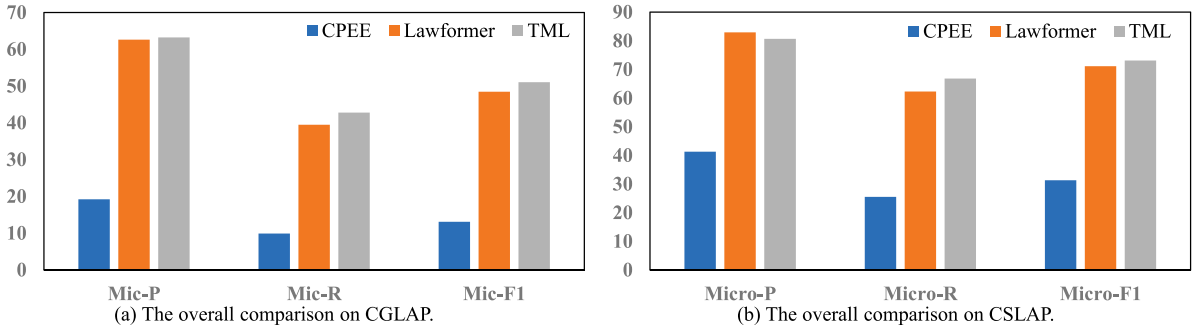
- Lawformer significantly surpasses all baselines. This indicates that the strong textual representation ability of the legal PLMs is beneficial for the CCP.
- Compared to Lawformer, TML further improves performance. Specifically, TML surpasses Lawformer 3.6%, and 5.3% in terms of accuracy and macro-F1, respectively. This evidence demonstrates that TML can better understand cases by adding special identifiers and perceived case structure to different legal elements, and by using multiple perspectives to handle the representation of different elements separately.

[7] https://pytorch.org/.
[8] https://github.com/thunlp/LegalPLMs.

Table 4
The main experimental results on CCP and FJP. The bold numbers indicate the best results.

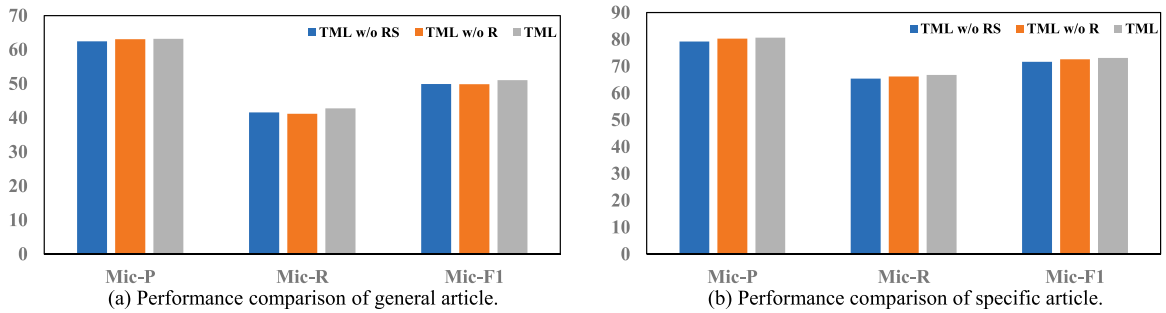
FJP					CCP				
Methods	Accuracy	Mac-P	Mac-R	Mac-F1	Methods	Accuracy	Mac-P	Mac-R	Mac-F1
SVM+word2vec	67.8	33.2	32.7	31.1	HMC	53.6	48.2	48.8	48.4
AutoJudge	75.1	41.7	33.4	28.7	Hdltex	47.4	7.9	16.7	10.7
MSJudge	66.1	32.9	39.9	34.4	HARNN	13.6	9.3	16.0	5.2
CCJudge	73.8	36.9	49.7	42.3	Flat-cause	50.0	25.0	33.3	27.8
BERT-Civil	73.5	41.8	34.1	30.2	Hie+fact	66.7	51.3	48.9	48.7
CPEE	**83.6**[a]	42.3	50.0	45.8	CPEE	70.6	55.6	64.0	58.9
Lawformer	77.4	71.0[a]	61.3[a]	64.8[a]	Lawformer	91.9[a]	89.8[a]	85.9[a]	87.3[a]
TML	81.4	**76.3**	**70.4**	**72.9**	TML	**95.5**	**93.5**	**92.1**	**92.6**

[a] The numbers represent the best performance of baseline.



(a) The overall comparison on CGLAP. (b) The overall comparison on CSLAP.

Fig. 5. The main experimental on CLAP. Subfigure (a) shows the performance comparison on CGLAP, and subfigure (b) indicates the performance comparison on CSLAP.



(a) Performance comparison of general article. (b) Performance comparison of specific article.

Fig. 6. The ablation study of CLAP. Subfigure (a) shows the performance comparison on CGLAP, and subfigure (b) indicates the performance comparison on CSLAP.

5.2.3. Comparison in terms of CLAP

We compare TML with the two most competitive baselines (*e.g.*, CPEE, and Lawformer), which are multi-task learning methods. From Fig. 5, we can draw the following findings.

- Lawformer exceeds CPEE in all metrics. Specifically, Lawformer beats CPEE in terms of micro-F1 by more than 30% on both CGLAP and CSLAP. This indicates that the strong representation ability of the legal PLMs can effectively solve the CLAP.
- Comparison to Lawformer, TML further improves performance. Specifically, TML exceeds Lawformer 2.6%, and 2.0% in terms of macro-F1 on CGLAP and CSLAP, respectively. This shows that the linguistic knowledge of corresponding causes is beneficial for predicting law articles.

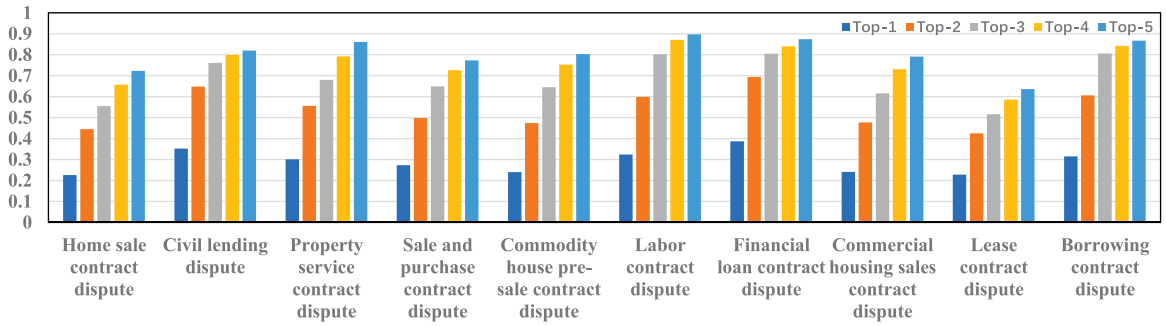
5.3. Ablation study (RQ2)

To answer RQ2, this section carries out an ablation study. To be specific, we have implemented two variants of TML: (1) TML w/o R that removes cause and article retrievers. (2) TML w/o RS that further removes structural perception operation on the basis of TML w/o R and uses global average pooling to obtain features. Table 5 illustrates the performance of these approaches on FJP and CCP, and Fig. 6 presents the comparison performance on CGLAP and CSLAP. We can conclude with the following findings.

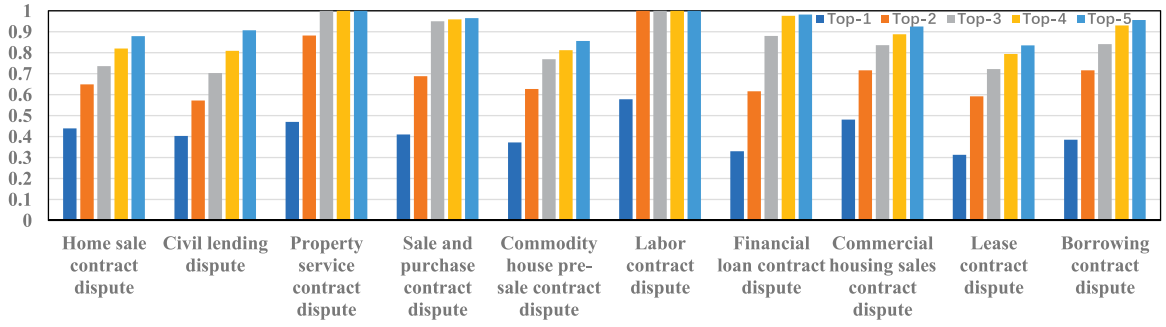
Table 5
The ablation study of FJP and CCP.

Methods	FJP				CCP			
	Accuracy	Mac-P	Mac-R	Mac-F1	Accuracy	Mac-P	Mac-R	Mac-F1
TML w/o RS	80.4	76.1	66.9	70.5	95.3	93.1	92.0	92.5
TML w/o R	80.7	74.3	**70.7**	72.3	95.2	92.8	**92.2**	92.4
TML	**81.4**	**76.3**	70.4	**72.9**	**95.5**	**93.5**	92.1	**92.6**

- On the FJP subtask, performance gradually decreases when retrievers and structural perception operation are gradually removed. This phenomenon demonstrates the effectiveness of the two designs for FJP.
- On the CCP subtask, TML acquires a performance boost when compared to TML w/o R. This indicates that the design of the retrievers does not compromise the performance of CCP while enhancing the FJP.
- On the CLAP subtask, comparing TML and TML w/o R, TML achieves the better performance in both specific and general articles. This indicates that the cause retriever is beneficial for law article prediction.



(a) Histogram of co-occurrence statistics for causes and general law articles.



(b) Histogram of co-occurrence statistics for causes and specific law articles.

Fig. 7. Histogram of co-occurrence statistics among causes and law articles. Subfigure (a) shows the co-occurrence statistics among causes and general articles, and subfigure (b) indicates the co-occurrence statistics among causes and specific articles.

5.4. Effectiveness of utilizing cause information (RQ3)

To answer RQ3, we carry out two quantitative analyses: (1) statistics of causes and law articles co-occurrence, and (2) quantitative analysis of law article prediction.

5.4.1. Statistics of cause and law article co-occurrence

Fig. 7 shows the histogram of co-occurrence statistics among causes and law articles. The horizontal coordinate is the cause, and the vertical coordinate is the percentage of co-occurrence of that cause and law articles. The different colored bars in the bar chart indicate the percentage of the number of co-occurrence of a particular cause and its top-k high-frequency law articles to the number of co-occurrences of that cause and all law articles. From this figure, we can find the following phenomena.

- The vast majority of causes and their top-5 high-frequency co-occurrence rate exceeded 80%. This phenomenon indicates that there is a correlation between cause and law articles.
- For all causes, the co-occurrence rate of the cause and the specific articles are greater than that of the general articles. This imagination indicates a greater correlation between the cause and specific legal provisions, possibly because certain specific legal provisions are specifically designed for specific cause.

5.4.2. Quantitative analysis of law article prediction

This subsection conducts the quantitative analysis of law article prediction to provide more insights. Specifically, we compare TML with Lawformer and divide the results of the law prediction into the following five situations.

- **Situation I:** The prediction labels are identical to the ground truth. In this case, the samples tend to be simple samples that the model handles well.
- **Situation II:** The prediction labels contain ground truth, and there is a multi-prediction error label in this situation.
- **Situation III:** The prediction labels are a subset of ground truth, in which the model predicts incompletely.
- **Situation IV:** The correctly predicted label in the prediction label is a subset of the ground truth, and there are redundant incorrect labels.
- **Situation V:** The predicted labels are all wrong. In this situation, the intersection of prediction labels and ground truth is empty. In this case, these samples are difficult to predict correctly for existing methods.

Table 6
The comparison of quantitative analysis of TML and Lawformer in terms of law article prediction results.

Tasks	CGLAP		CSLAP	
Methods	Lawformer	TML	Lawformer	TML
Situation I	20.6	22.5	51.6	53.8
Situation II	7.4	8.0	4.0	5.7
Situation III	17.0	18.5	16.4	16.1
Situation IV	11.0	10.7	2.1	2.8
Situation V	44.1	40.3	25.9	21.7

The Table 6 illustrates the count the percentage of different scenarios in the test set. The following findings can be observed.

- In CGLAP, the prediction results of TML are 22.5% and 40.3% in situation I and situation V, respectively. In CSLAP, the prediction results of TML are 53.8% and 21.7% in situation I and situation V, respectively. This phenomenon indicates: (1) there are still many difficult samples that TML cannot handle in general and special law articles; (2) The prediction of the general law articles is more difficult than that of the specific law articles.
- In situation I, TML outperforms Lawformer 1.9%, 2.2% in terms of CGLAP and CSLAP, respectively. This reflects that our method is effective in improving the law prediction.
- In situation V, TML is 3.8%, 4.2% below Lawformer in terms of CGLAP and CSLAP, respectively. In situation II, TML is 0.6%, 1.7% outperform Lawformer regarding CGLAP and CSLAP. For cases where the prediction is partially correct (situation III and situation IV), TML outperforms Lawformer by 1.2% and 0.4% in terms of CGLAP and

Table 7
The quantitative analysis of TML and Lawformer on FJP.

Situation	Lawformer	TML	Count	Ratio
Situation I	✓	✓	22,717	70.9
Situation II	✓	✗	2,087	6.5
Situation III	✗	✓	3,358	10.5
Situation IV	✗	✗	3,862	12.1

CSLAP. This phenomenon suggests that our method can mitigate the extent of some of the difficult sample errors.

5.5. Effectiveness of utilizing law articles (RQ4)

The quantitative and qualitative analyses are conducted to answer RQ4. Firstly, we quantitatively analyze the performance of TML and Lawformer on FJP. Secondly, we select several samples where Lawformer predicts incorrectly but TML predicts correctly to perform the case study.

5.5.1. Quantitative analysis

This subsection conducts the quantitative analysis of TML and Lawformer on FJP. The following four situations are discussed.

- **Situation I.** Both TML and Lawformer predict correctly. The samples in that situation can often be considered as simple samples that the model can handle.
- **Situation II.** The TML predicts incorrectly while Lawformer predicts correctly. The situation reflects the failure of our method with respect to Lawformer.
- **Situation III.** The Lawformer predicts error, while TML predicts correctly. This situation reflects the ability of our model to handle samples that are difficult for Lawformer to handle.
- **Situation IV.** Both TML and Lawformer predict incorrectly. These samples can be considered as difficult samples that Lawformer and TML are currently unable to handle.

Table 7 illustrates the experimental results of quantitative analysis on FJP. We report the number and percentage of different situations in terms of the test set. The following findings can be drawn from this table.

- The percentage of situation I is 70.9%. It indicates that the legal PLMs perform well on FJP, and most cases can be solved.
- To our surprise, the percentage of situation II is 6.5%. This indicates that our method fails on some samples compared to Lawformer. The possible reason is that the current poor performance of law articles prediction leads to the wrong candidate obtained by the article retrieval mechanisms, and this error further affects the prediction of FJP.
- The percentage of situation III is 10.5%. This shows that TML can effectively utilize the law information to improve the performance of FJP. The percentage of situation III is greater than that of II. This indicates that the improvement obtained by correct article retrieval is great than the case of performance loss caused by wrong article search.
- The percentage of situation IV is 12.1%. This indicates that there are still a number of difficult cases that are difficult to solve for the Lawformer and TML. One possible reason is that the prediction error of CLAP affects the law retrieval to the correct candidate articles, which further affects the FJP.

5.5.2. Case study

To better investigate how law articles boost TML, we conduct qualitative analysis for FJP. To achieve this goal, we demonstrate two illustrative examples to conduct the case study. Fig. 8 illustrates two simplified civil cases. The left half contains civil case information. The upper right part shows the law articles prediction of TML, and the lower right part shows the final judgment prediction of Lawformer and TML respectively.

From Fig. 8(a), the case shows that the defendant only pays a portion of the payment within the agreed period, and the plaintiff fails to collect the outstanding payment. The plaintiff's plea is for payment of the outstanding and interest for delayed performance. The ground truth of this plea is *supported pleas*, and Lawformer is incorrectly predicted to be *partially supported pleas*. Our model correctly predicts into *supported pleas* probably by utilizing the prediction law articles information. The *Article 159* and *Article 161* stipulate that person shall pay the price in the agreed amount and at the time within the agreed period respectively. The *Article 107* stipulates that the person performing the contract, who does not comply with the provisions of the contract, shall be liable for breach of contract, such as compensation for damages. Combining this law articles' information, TML may understand that the defendant has failed to make the required payments within the agreed period and therefore needs to compensate for the damages caused by the breach of contract, thus making a correct prediction.

From Fig. 8(b), the case describes that the defendant borrows money from the plaintiff and repays only part of the amount owed. The plaintiff's plea is for the return of the remaining principal amount and interest. The ground truth of this plea is *supported pleas*, and Lawformer is incorrectly predicted to be *partially supported pleas*. The *Article 207* stipulates that the borrower should pay late interest when they is late. By utilizing the corresponding articles' information, our model comes up with the correct prediction.

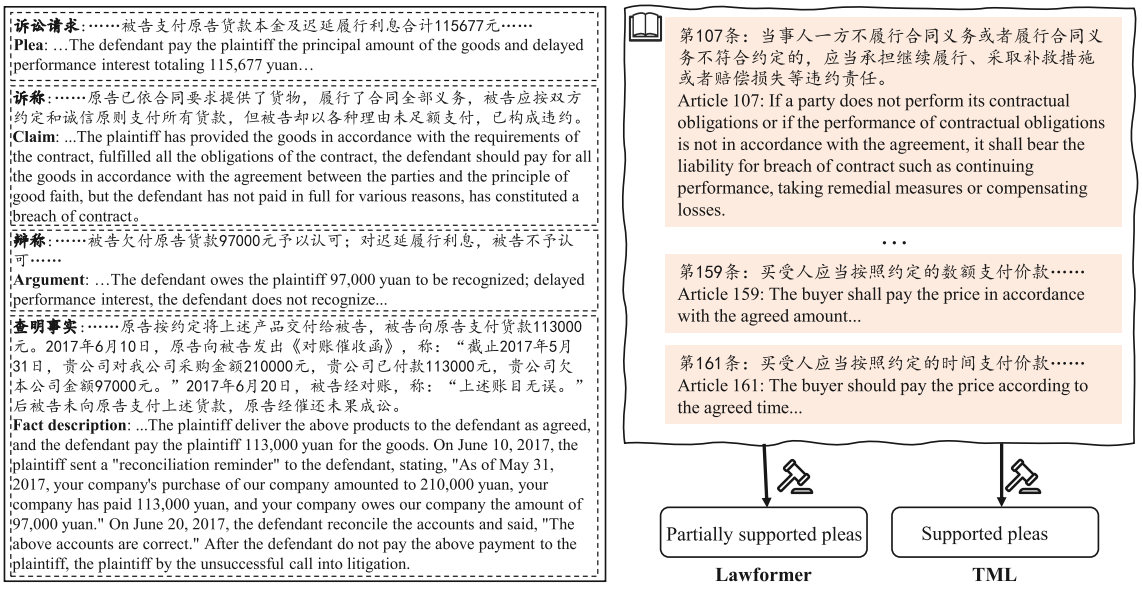
5.6. Performance on low-resource scenarios (RQ5)

To further explore the capabilities of the TML, we conduct experiments with low-resource setting. Specifically, we sample the civil cases to ensure that there are only 50 cases per cause (50-shot), a total of 500 civil cases. For a fair comparison, the Lawformer is selected as baseline, since our method is also Lawformer-based. To avoid random results due to the small sample size, we use three different sets of randomly seed sampled cases to generate three different training sets. Table 8 presents the comparison results between FJP and CCP. Table 9 shows the comparison results in terms of CGLSP and CSLAP. The following experimental phenomena can be observed.

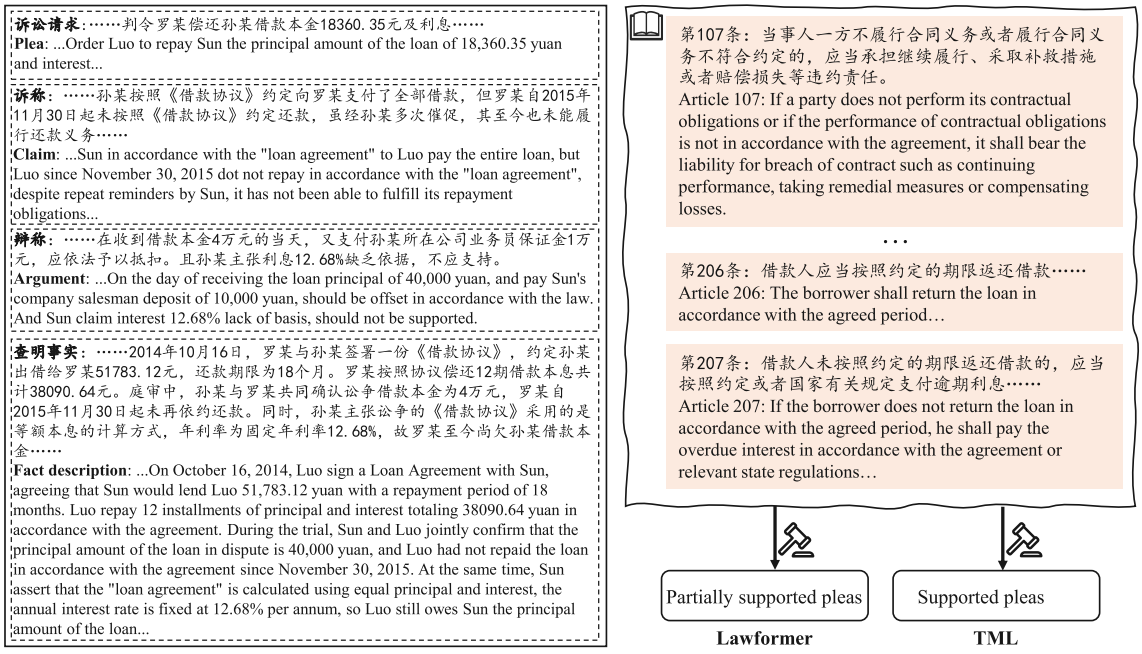
- On the FJP, the accuracy of both methods is substantially higher than macro-F1. The reason for this phenomenon may be that the model tends to overfit at low-resource settings, thus biasing the prediction into certain classes. The FJP test set is an imbalance dataset, and thus the accuracy is significantly higher than that of macro-F1.
- On the FJP, TML is 18% higher than Lawformer in the macro-F1, indicating that TML can effectively mitigate the overfitting problem in the low-resource scenario by introducing external knowledge of the law articles.
- On the CCP, TML outperforms Lawformer in all metrics. It indicates that the case-structure-aware multi-view feature design remains effective in low-resource scenarios.
- Our method exceeds Lawformer by 6.9% and 19.8% micro-F1 on CGLAP and CSLAP respectively. This means that the performance of law prediction can be effectively promoted by exploiting the linguistic knowledge of the cases in the absence of sufficient training samples.

It is interesting to explore the reasons why Lawformer and TML have such low F1 in low-resource situations. Consequently, we compare the performance of these two methods under each class (see Table 10). We can draw the following conclusions:

- Due to the lack of sufficient training samples, the two methods suffer from overfitting and thus are biased to make predictions that favor the *supported pleas* category, while their performance in predicting *partially supported pleas*, and *rejected pleas* categories is markedly poor.



(a) The ground truth is *supported pleas*. Lawformer incorrectly predicts into *partially supported pleas*, while TML predicts correctly.



(b) The ground truth is *supported pleas*. Lawformer incorrectly predicts into *rejected pleas*, while TML predicts correctly.

Fig. 8. Case study of two simplified cases from test set.

- For the cases with ground truth are *partially supported pleas*, *rejected pleas*, Lawformer predicts almost all to *supported pleas*. This bias leads to a very low macro-F1. We argue that most baselines in Table 4 also encounter this problem.
- TML improves the prediction ability of *partially supported pleas*, *rejected pleas* with the guarantee of *supported pleas* performance, thus improving the performance from the whole.

6. Ethical discussion

This section discusses the ethical concerns of TML: 1) TML aims to utilize AI technology to assist judges, rather than replace legal practitioners. The authority to make decisions in civil cases should firmly rest with the judge. 2) CCJP is an emerging technology, and existing models may not understand the corresponding legal theories. Thus, users should understand these models' limitations to prevent potential misuse.

7. Conclusions and further directions

We present a novel topology-aware multi-task learning framework, called TML, which exploits more comprehensive legal essential elements and possesses the ability to handle numerous causes. Concretely, TML utilizes legal PTMs as its backbone and incorporates specific tokens for legal elements, allowing for the perception of structural information within the case. In addition, We develop three parameter-free retrieval mechanisms that enable the injection of topological dependencies between various subtasks. Extensive experiments are conducted

Table 8
The comparison results of FJP and CCP in low-resource setting. The average (±std) represents the average (standard deviation) of the results of different random train sets.

Models	Seeds	FLP				CCP			
		Accuracy	Mac-P	Mac-R	Mac-F1	Accuracy	Mac-P	Mac-R	Mac-F1
Lawformer	666	0.682	0.227	0.333	0.270	0.795	0.622	0.682	0.639
	777	0.682	0.329	0.334	0.272	0.788	0.629	0.674	0.638
	888	0.682	0.227	0.333	0.270	0.764	0.714	0.688	0.647
	Average	0.682	0.261	0.333	0.271	0.783	0.655	0.681	0.641
	±std	±0.000	±0.048	±0.000	±0.001	±0.013	±0.042	±0.006	±0.004
TML	666	0.698	0.568	0.465	0.481	0.827	0.738	0.758	0.742
	777	0.693	0.566	0.406	0.411	0.812	0.732	0.750	0.730
	888	0.700	0.604	0.440	0.461	0.772	0.728	0.718	0.660
	Average	0.697	0.580	0.437	0.451	0.803	0.733	0.742	0.711
	±std	±0.003	±0.018	±0.024	±0.029	±0.023	±0.004	±0.017	±0.037

Table 9
The comparison results of CGLAP and CSLAP in low-resource setting. The average (±std) represents the average (standard deviation) of the results of different random train sets.

Model	Seeds	CGLAP			CSLAP		
		Mic-P	Mic-R	Mic-F1	Mic-P	Mic-R	Mic-F1
Lawformer	666	0.469	0.255	0.330	0.445	0.268	0.334
	777	0.475	0.178	0.259	0.445	0.268	0.335
	888	0.469	0.482	0.475	0.445	0.268	0.335
	Average	0.471	0.305	0.355	0.445	0.268	0.335
	±std	±0.003	±0.129	±0.090	±0.000	±0.000	±0.000
TML	666	0.590	0.358	0.445	0.808	0.584	0.678
	777	0.575	0.326	0.417	0.849	0.412	0.555
	888	0.525	0.337	0.411	0.793	0.238	0.366
	Average	0.564	0.340	0.424	0.817	0.411	0.533
	±std	±0.028	±0.013	±0.015	±0.024	±0.141	±0.128

Table 10
The comparison results on FJP in label-level.

Labels of FJP	Number	Lawformer			TML		
	of cases	P	R	F1	P	R	F1
Rejected	6,469	0.306	0.003	0.007	0.537	0.196	0.287
Partially supported	3,707	0.000	0.000	0.000	0.450	0.080	0.136
Supported	21,848	0.683	0.998	0.811	0.711	0.944	0.811
Macro-average	32,024	0.329	0.334	0.273	0.566	0.407	0.411

on a publicly available real dataset, and experimental results show the superiority of TML.

Despite making impressive strides in the field, some unresolved challenges remain. Firstly, both FJP and CLAP tasks have room for improvement in overall performance. The CLAP is an extremely multi-label classification problem. Certain difficult cases in the FJP may require a more in-depth understanding of the case material or a combination of more precise external knowledge. Secondly, there is greater potential to enhance our method in low-resource settings. We argue that such settings have practical implications in situations where large-scale data is difficult to obtain in sensitive legal areas. Thirdly, the current approaches lack interpretability and robustness, which promote reliability and trust in the judgment results. It is extremely important in the sensitive area of law. In the future, we plan to explore the use of larger-scale PTMs (Cui et al., 2023; Cui, Yang and Yao, 2023; Huang et al., 2023) to address these issues. Meanwhile, in order to better utilize PTMs that require high computational power, we plan to explore methods for accelerating PTMs, such as system scheduling (Li, Tang, & Li, 2013), training acceleration (Aminabadi et al., 2022), and application in supercomputers (Xiao et al., 2019).

CRediT authorship contribution statement

Yuquan Le: Conceptualization, Methodology, Validation, Writing – original draft. **Sheng Xiao:** Supervision, Writing – review & editing, Funding acquisition. **Zheng Xiao:** Supervision, Writing – review & editing, Funding acquisition. **Kenli Li:** Supervision, Review, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

I have shared the link to my data in the paper.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (2022YFC3303400, 2022YFC3301500).

References

Aminabadi, R. Y., Rajbhandari, S., Awan, A. A., Li, C., Li, D., Zheng, E., et al. (2022). DeepSpeed-inference: Enabling efficient inference of transformer models at unprecedented scale. In *2022 SC22: International conference for high performance computing, networking, storage and analysis (SC)* (pp. 1–15).

Ashley, K. D. (2019). A brief history of the changing roles of case prediction in AI and law. *Law in Context. A Socio-legal Journal*, *36*, 93.

Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The long-document transformer. ArXiv preprint, abs/2004.05150.

Bi, S., Zhou, Z., Pan, L., & Qi, G. (2022). Judicial knowledge-enhanced magnitude-aware reasoning for numerical legal judgment prediction. *Artificial Intelligence and Law*, 1–34.

Branting, L. K., Pfeifer, C., Brown, B., Ferro, L., Aberdeen, J., Weiss, B., et al. (2021). Scalable and explainable legal prediction. *Artificial Intelligence and Law*, *29*, 213–238.

Cardellino, C., Teruel, M., Alemany, L. A., & Villata, S. (2017). A low-cost, high-coverage legal named entity recognizer, classifier and linker. In *Proceedings of the 16th edition of the international conference on articial intelligence and law* (pp. 9–18).

Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., & Androutsopoulos, I. (2020). LEGAL-BERT: The muppets straight out of law school. In *Findings of the association for computational linguistics: EMNLP 2020* (pp. 2898–2904).

Chen, H., Cai, D., Dai, W., Dai, Z., & Ding, Y. (2019). Charge-based prison term prediction with deep gating network. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing* (pp. 6362–6367).

Chen, C., Li, K., Teo, S. G., Zou, X., Li, K., & Zeng, Z. (2020). Citywide traffic flow prediction based on multiple gated spatio-temporal convolutional neural networks. *ACM Transactions on Knowledge Discovery from Data*, *14*(4), 1–23.

Chopra, S., Hadsell, R., & LeCun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE computer society conference on computer vision and pattern recognition, Vol. 1* (pp. 539–546).

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*, 273–297.

Cui, G., Li, W., Ding, N., Huang, L., Liu, Z., & Sun, M. (2023). *Decoder tuning: Efficient language understanding as decoding*. The Association for Computational Linguistics.

Cui, Y., Yang, Z., & Yao, X. (2023). Efficient and effective text encoding for Chinese llama and alpaca. ArXiv preprint, abs/2304.08177.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (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* (pp. 4171–4186).

Feng, Y., Li, C., & Ng, V. (2022). Legal judgment prediction: A survey of the state of the art. In *Proceedings of the thirty-first international joint conference on artificial intelligence* (pp. 5461–5469).

Gan, L., Kuang, K., Yang, Y., & Wu, F. (2021). Judgment prediction via injecting legal knowledge into neural networks. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 35* (pp. 12866–12874).

Hu, Z., Li, X., Tu, C., Liu, Z., & Sun, M. (2018). Few-shot charge prediction with discriminative legal attributes. In *Proceedings of the 27th international conference on computational linguistics* (pp. 487–498).

Huang, Q., Tao, M., An, Z., Zhang, C., Jiang, C., Chen, Z., et al. (2023). Lawyer llama technical report. ArXiv, abs/2305.15062.

Kanapala, A., Pal, S., & Pamula, R. (2019). Text summarization from legal documents: a survey. *Artificial Intelligence Review*, *51*, 371–402.

Katz, D. M., Bommarito, M. J., & Blackman, J. (2017). A general approach for predicting the behavior of the supreme court of the United States. *PLoS One*, *12*(4), Article e0174698.

Le, Y., He, C., Chen, M., Wu, Y., He, X., & Zhou, B. (2020). Learning to predict charges for legal judgment via self-attentive capsule network. In *European conference on artificial intelligence* (pp. 1802–1809).

Le, Y., Zhao, Y., Chen, M., Quan, Z., He, X., & Li, K. (2022). Legal charge prediction via bilinear attention network. In *Proceedings of the 31st ACM international conference on information & knowledge management* (pp. 1024–1033).

Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2022). A survey of convolutional neural networks: Analysis, applications, and prospects.. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(12), 6999–7019.

Li, K., Tang, X., & Li, K. (2013). Energy-efficient stochastic task scheduling on heterogeneous computing systems. *IEEE Transactions on Parallel and Distributed Systems*, *25*(11), 2867–2876.

Limsopatham, N. (2021). Effectively leveraging BERT for legal document classification. In *Proceedings of the natural legal language processing workshop 2021* (pp. 210–216).

Liu, C., Li, K., Li, K., & Buyya, R. (2017). A new service mechanism for profit optimizations of a cloud provider and its users. *IEEE Transactions on Cloud Computing*, *9*(1), 14–26.

Long, S., Tu, C., Liu, Z., & Sun, M. (2019). Automatic judgment prediction via legal reading comprehension. In *China national conference on chinese computational linguistics* (pp. 558–572).

Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Ma, L., Zhang, Y., Wang, T., Liu, X., Ye, W., Sun, C., et al. (2021). Legal judgment prediction with multi-stage case representation learning in the real court setting. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval* (pp. 993–1002).

Medvedeva, M. (2022). Identification, categorisation and forecasting of court decisions.

Medvedeva, M., Wieling, M., & Vols, M. (2023). Rethinking the field of automatic prediction of court decisions. *Artificial Intelligence and Law*, *31*(1), 195–212.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th international conference on neural information processing systems* (pp. 3111–3119).

Păi, V., Mitrofan, M., Gasan, C. L., Coneschi, V., & Ianov, A. (2021). Named entity recognition in the Romanian legal domain. In *Proceedings of the natural legal language processing workshop 2021* (pp. 9–18).

Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., & Huang, X. (2020). Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, *63*(10), 1872–1897.

Turtle, H. (1995). Text retrieval in the legal world. *Artificial Intelligence and Law*, *3*, 5–54.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. In *Proceedings of the 31st international conference on neural information processing systems* (pp. 6000–6010).

Wehnert, S., Dureja, S., Kutty, L., Sudhi, V., & De Luca, E. W. (2022). Applying BERT embeddings to predict legal textual entailment. *The Review of Socionetwork Strategies*, *16*(1), 197–219.

Weng, T., Zhou, X., Li, K., Tan, K.-L., & Li, K. (2022). Distributed approaches to butterfly analysis on large dynamic bipartite graphs. *IEEE Transactions on Parallel and Distributed Systems*, *34*(2), 431–445.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., et al. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations* (pp. 38–45).

Xiao, C., Hu, X., Liu, Z., Tu, C., & Sun, M. (2021). Lawformer: A pre-trained language model for chinese legal long documents. *AI Open*, *2*, 79–84.

Xiao, G., Li, K., Chen, Y., He, W., Zomaya, A. Y., & Li, T. (2019). Caspmv: a customized and accelerative SPMV framework for the sunway TaihuLight. *IEEE Transactions on Parallel and Distributed Systems*, *32*(1), 131–146.

Xiong, C., Zhong, V., & Socher, R. (2016). Dynamic coattention networks for question answering. In *International Conference on Learning Representations*.

Yang, Y., Wang, X., Zhou, D., Wei, D.-Q., & Peng, S. (2022). Svpath: an accurate pipeline for predicting the pathogenicity of human exon structural variants. *Briefings in Bioinformatics*, *23*(2).

Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: LSTM cells and network architectures. *Neural Computation*, *31*(7), 1235–1270.

Yu, W., Sun, Z., Xu, J., Dong, Z., Chen, X., Xu, H., et al. (2022). Explainable legal case matching via inverse optimal transport-based rationale extraction. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval* (pp. 657–668).

Zhang, H., Wang, X., Tan, H., & Li, R. (2019). Applying data discretization to DPCNN for law article prediction. In *Natural language processing and chinese computing* (pp. 459–470).

Zhao, J., Guan, Z., Xu, C., Zhao, W., & Chen, E. (2022). Charge prediction by constitutive elements matching of crimes. In *Proceedings of the thirty-first international joint conference on artificial intelligence, Vol. 22* (pp. 4517–4523).

Zhao, L., Yue, L., An, Y., Liu, Y., Zhang, K., He, W., et al. (2021). Legal judgment prediction with multiple perspectives on civil cases. In *Artificial intelligence: First CAAI international conference* (pp. 712–723).

Zhao, L., Yue, L., An, Y., Zhang, Y., Yu, J., Liu, Q., et al. (2022). CPEE: Civil case judgment prediction centering on the trial mode of essential elements. In *Proceedings of the 31st ACM international conference on information & knowledge management* (pp. 2691–2700).

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., et al. (2023). A survey of large language models. ArXiv preprint, abs/2303.18223.

Zhong, H., Xiao, C., Tu, C., Zhang, T., Liu, Z., & Sun, M. (2020). How does NLP benefit legal system: A summary of legal artificial intelligence. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 5218–5230).

Zhong, K., Yang, Z., Xiao, G., Li, X., Yang, W., & Li, K. (2021). An efficient parallel reinforcement learning approach to cross-layer defense mechanism in industrial control systems. *IEEE Transactions on Parallel and Distributed Systems*, *33*(11), 2979–2990.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., et al. (2020). Graph neural networks: A review of methods and applications. *AI Open*, *1*, 57–81.

Zhou, X., Zhang, Y., Liu, X., Sun, C., & Si, L. (2019). Legal intelligence for e-commerce: multi-task learning by leveraging multiview dispute representation. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval* (pp. 315–324).

Zou, B. (2010). The nine steps of trial of essential items.