ML-LJP: Multi-Law Aware Legal Judgment Prediction

Yifei Liu*
School of Software Technology, Zhejiang University
Hangzhou, Zhejiang, China
liuyifei@zju.edu.cn

Yiquan Wu*†
College of Computer Science and Technology, Zhejiang University
Hangzhou, Zhejiang, China
wuyiquan@zju.edu.cn

Yating Zhang
Alibaba Group
Hangzhou, Zhejiang, China
yatingz89@gmail.com

Changlong Sun
Alibaba Group
College of Computer Science and Technology, Zhejiang University
Hangzhou, Zhejiang, China
changlong.scl@taobao.com

Weiming Lu
College of Computer Science and Technology, Zhejiang University
Hangzhou, Zhejiang, China
luwm@zju.edu.cn

Fei Wu
College of Computer Science and Technology, Zhejiang University
Hangzhou, Zhejiang, China
wufei@cs.zju.edu.cn

Kun Kuang†
College of Computer Science and Technology, Zhejiang University
Hangzhou, Zhejiang, China
kunkuang@zju.edu.cn

ABSTRACT

Legal judgment prediction (LJP) is a significant task in legal intelligence, which aims to assist the judges and determine the judgment result based on the case's fact description. The judgment result consists of law articles, charge, and prison term. The law articles serve as the basis for the charge and the prison term, which can be divided into two types, named as charge-related law article and term-related law article, respectively. Recently, many methods have been proposed and made tremendous progress in LJP. However, the existing methods only focus on the prediction of the charge-related law articles, ignoring the term-related law articles (e.g., laws about lenient treatment), which limits the performance in the prison term prediction. In this paper, following the actual legal process, we expand the law article prediction as a multi-label classification task that includes both the charge-related law articles and term-related law articles and propose a novel multi-law aware LJP (ML-LJP) method to improve the performance of LJP. Given the case's fact description, firstly, the label (e.g., law article and charge) definitions in the Code of Law are used to transform the representation of the fact into several label-specific representations and make the prediction of the law articles and the charge. To distinguish the similar content of different label definitions, contrastive learning is conducted in the training. Then, a graph attention network (GAT) is applied to learn the interactions among the multiple law articles for the prediction of the prison term. Since numbers (e.g., amount of theft and weight of drugs) are important for LJP but often ignored by conventional encoders, we design a corresponding number representation method to locate and better represent these effective numbers. Extensive experiments on real-world dataset show that our method achieves the best results compared to the state-of-the-art models, especially in the task of prison term prediction where ML-LJP achieves a 10.07% relative improvement over the best baseline.

CCS CONCEPTS

• **Applied computing** → **Law**; • **Computing methodologies** → **Supervised learning by classification**.

KEYWORDS

LegalAI; Multi-task Learning; Graph Attention Network

ACM Reference Format:
Yifei Liu, Yiquan Wu, Yating Zhang, Changlong Sun, Weiming Lu, Fei Wu, and Kun Kuang. 2023. ML-LJP: Multi-Law Aware Legal Judgment Prediction. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), July 23–27, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3539618.3591731

*Equal contribution.
†Corresponding author.

SIGIR '23, July 23–27, 2023, Taipei, Taiwan.

ACM ISBN 978-1-4503-9408-6/23/07…$15.00
https://doi.org/10.1145/3539618.3591731

1 INTRODUCTION

Recently, NLP techniques have been widely applied in Legal Artificial Intelligence (LegalAI) and significantly improve the effectiveness and efficiency of a judge from different aspects, such as legal judgment prediction [11, 60], court view generation, [46] and similar case matching [2]. As an important component of the judicial process, legal judgment prediction (LJP) is one of the most challenging research topics in LegalAI [60]. LJP aims to predict the judgment results (e.g., law article, charge, and prison term) of legal cases according to the case's fact descriptions.

The law articles are the basis for the determination of the charge and the prison term, and there are two types of law articles, named charge-related law articles and term-related law articles, respectively. As shown in Fig. 1, the judgment of the example case involves three law articles, including one charge-related law article (e.g., *Article 347*) and two term-related law articles (e.g., *Article 65* and *Article 67*). However, despite achieving tremendous progress, previous LJP methods take only charge-related law articles into consideration and ignore the prediction of the term-related law articles, which departs from the real judicial scenarios that most of the cases involve both two types of law articles. The ignorance of term-related law articles also limits the performance of prison term prediction, which is the downstream task in LJP. In the state-of-the-art models, the accuracy of the charge and the article prediction is more than 90%, but the accuracy of the prison term prediction is under 45% [51, 56]. Therefore, taking the term-related law articles into consideration is a sensible idea.

To accomplish the idea, we still face two challenges: **1) The similar definitions of law articles in the Code of Law.** Different law articles have different legal effects, even for similar law articles, so the model should clearly distinguish them before making judgment predictions. **2) The complex interactions among the law articles of one case.** For example, in Fig. 1, the charge-related *Article 347* will decide the rough range of the prison term. The term-related *Article 65* (recidivism) will increase the prison term but *Article 67* (confession) will reduce it. To better predict the prison term, the model should learn the complex interaction.

In this paper, we expand the law article prediction as a multi-label classification task that includes both charge-related and term-related law articles and propose a multi-law aware legal judgment prediction (ML-LJP) method to improve the LJP performance. Given the case's fact description, we first encode it several times since the text semantics are sophisticated. Specifically, each time we use a label (e.g., law article, charge) as the query and do an attention operation on the fact to get the label-specific (e.g., law-specific and charge-specific) fact representation. Instead of randomly initializing the query like [49], we utilize the label definition in the Code of Law as the query. To distinguish the similar definitions, we conduct contrastive learning in the training. The charge and law articles are predicted by the label-specific fact representations. Then, given the corresponding law-specific fact representations of predicted law articles, a graph attention network (GAT) is applied to learn the high-order interactions among multiple law articles and output the aggregated fact representation to enhance the performance of prison term prediction. Moreover, since some numbers (e.g., amount of theft and weight of drugs) are important but often ignored by conventional encoders, we design a simple yet effective number representation method to locate and enhance numbers.

Most of the existing LJP works conducted experiments on the CAIL2018 dataset [48]. However, in the CAIL2018 dataset, only the charge-related law articles are labeled, so the judgments are not complete. Therefore, we conduct the experiments on the newly released LAIC2021[1] dataset, which reserves all the law articles in the judgment. [2] The experimental results show that our ML-LJP has a better performance compared to the state-of-the-art models, especially in prison term prediction. Further studies demonstrate that the ML-LJP can accurately extract label-specific parts from fact descriptions and capture the interactions between multiple law articles. The analysis study also shows the effectiveness of the proposed numerical representation method.

To summarize, our major contributions are listed as follows:

- We explore the legal judgment prediction (LJP) task by taking all types of law articles into consideration, including charge-related law articles and term-related articles.
- We propose a novel multi-law aware LJP (ML-LJP) method to improve the LJP by extracting the label-specific features of the fact and capturing the high-order interactions among multiple law articles.
- The experiments on a real-world legal document dataset LAIC2021 show that our method achieves the best performance compared to SOTA models, especially on prison term prediction where ML-LJP achieves a 10.07% relative improvement over the best baseline.
- To help the reproducibility of the proposed method, we make all the data and code publicly available[3].

2 RELATED WORK

2.1 Legal Artificial Intelligence.

Legal Artificial Intelligence (LegalAI) aims to apply artificial intelligence techniques to the legal domain. Legal jobs often involve plenty of paperwork, which requires legal professionals a lot of time and effort. Recently, a number of NLP techniques have been proposed and applied in the legal domain to assist legal professionals in some textual work, such as Legal Judgment Prediction [47, 51, 60, 63], Court View Generation [46], Similar Case Matching, [2] and Legal Question Answering [61]. In this paper, we focus on legal judgment prediction.

2.2 Legal Judgment Prediction.

Legal Judgment Prediction (LJP) plays a significant role in LegalAI. Given the fact description, LJP contains the prediction of law articles, charge, and prison term. Early LJP works mainly relied on rule-based or mathematical methods [21, 33, 37], which require lots of manually extracted features. The rule-based LJP methods are accurate but difficult to generalize because of the high cost of feature definition. In recent years, deep learning has been proven to be effective in many domains [17, 22, 28, 34, 41, 45, 53, 62, 64], since deep learning requires far less labor, many researchers begin to explore LJP using deep learning technology. Most LJP methods consider the three subtasks as three single-label classification tasks. Zhong et al. [60] model the dependency of three subtasks by topological learning. Yue et al. [56] splits the fact description into different parts for predictions. Xu et al. [51] uses a graph distillation to extract discriminative features from labels. Feng et al.

[1]Dataset can be downloaded from http://data.court.gov.cn/pages/laic2021.html
[2]Following the previous work, we divide the prison term into several non-overlapping intervals.
[3]https://github.com/6666ev/ML-LJP

Fact Description	After the trial, it was found that on March 17, 2015, at about 23:00 pm, the defendant A sold 1.5 grams of methamphetamine to B for RMB 500 yuan. At about 17:00 on the afternoon of 22nd of the same month, the defendant A sold 1.5 grams of methamphet-amine to B for RMB 650 yuan. On the evening of 23rd of the same month, the defendant A was caught by … and 0.8351 grams of methamphetamine was seized from his possession ... During the trial, the defendant A withdrew the stolen money of RMB 1,150 yuan. The defendant A was sentenced to one year and six months imprisonment … for the crime of harboring prostitution … The defendant A is a recidivist and should be punished severely. After his return to the court, the defendant A was able to confess the facts of his crime truthfully, so he could be punished lightly.		
Judgment	Law Articles	Charge	Prison Term
	Article 347, 65, 67	Drug Trafficking	12 months

Article 347: Smuggling, trafficking, transportation, manufacturing of drugs, regardless of the amount, should be held criminally responsible and criminaly punishable.

Article 65: If a criminal sentenced to fixed-term imprisonment commits the crime that should be sentenced to fixed-term imprisonment again within five years, he is a recidivist and should be …

Article 67: For criminals who voluntarily surrenders to the police after committing a crime and truthfully confesses his crime, the punishment can be mitigated.

Drug Trafficking: The crime is committed intentionally and directly, e.g., knowingly smuggling, trafficking, transporting, or manufacturing drugs, but negligence does not constitute the crime…

Figure 1: An example of a drug trafficking case. Given the fact description, the judge will determine the law articles, the charge and the prison term. The right part is the corresponding label (e.g., law article and charge) definitions in the Code of Law. *Article 347* is the charge-related, and *Article 65* and *Article 67* are the term-related.

[10] leverages annotated legal event extraction dataset to locate key event in criminal cases. Liu et al. [24], Zhang et al. [57] use contrastive learning to capture fine-grained differences between similar articles or charges for the LJP task.

Research about LJP has been conducted in many countries, but these LJP methods may not be universally applicable due to the varying legal systems in different countries [30–32]. In this paper, we focus on the LJP in Chinese legal system. Different from previous methods, we expand the law article prediction as a multi-label classification task that includes both charge-related law articles and term-related law articles and learn the interactions among the predicted law articles to improve the performance.

2.3 Other Related Techniques.

Here we introduce three other related techniques:

• **Contrastive Learning** has been a promising trend in discriminative representation learning [1, 4, 15]. The general idea is to pull an instance closer to its positive samples and push away its negative samples in the embedding space. Gao et al. [12], Yan et al. [52] attempt to improve sentence embedding via contrastive learning. Zhang et al. [59] utilizes contrastive learning and introduces label meta-data for zero-shot multi-label text classification. In this paper, we use contrastive learning to distinguish the similar content of different label definitions.

• **Graph Neural Network** (GNN) has attracted extensive attention due to its significant performance in modeling graph structure data [14, 19, 39, 44]. GCN [19] adopts a convolution operator over the graph for information propagation. GAT [39] specifies different weights to different nodes in a neighborhood by attention mechanism. Lin et al. [23], Yao et al. [55] propose to learn relationships between texts and words by GCN for text classification. In the task of multi-label text classification, Lu et al. [25] aggregates multiple label graphs and [27] employs dual GCN to model adaptive interactions among label-specific components. In this paper, we use GAT to learn the interactions among the multiple law articles.

• **Number Representation** has recently drawn attention because numbers are an important part of texts, while the previous NLP research has generally ignored them. Thawani et al. [36] divides the existing methods into string-based methods [13, 40, 58] and real-based methods [35, 40]. In this paper, We mainly follow the previous work [16, 58] that converts a number into the scientific notation and encodes the mantissa and exponent part of the number respectively. We also take advantage of the unit, which is an important part of the number in legal scenarios.

3 METHODOLOGY

In this section, we describe our multi-law based LJP (ML-LJP) model, Fig. 2 shows the model architecture. Firstly, we define some notations in Tab. 1 and formulate LJP task as:

PROBLEM 1 (LEGAL JUDGMENT PREDICTION). *Given fact description f of a case, our goal is to predict the law articles a, the charge c, and the prison term p.*

Notation	Description
$f = \{w_1^f, ..., w_{l_f}^f\}$	A word sequence of fact description.
(a, c, p)	The (law articles, charge and prison term) in the judgment.
$x^{a_i} = \{w_1^{a_i}, ..., w_{l_a}^{a_i}\}$	A word sequence of law article a_i definition.
$x^{c_j} = \{w_1^{c_j}, ..., w_{l_c}^{c_j}\}$	A word sequence of charge c_j definition.

Table 1: Notations.

3.1 Fact Description Representation

In this section, we first introduce the number representation method, then we introduce how to get label-specific fact representation.

3.1.1 ***Number Representation.*** The legal judgment is sensitive to the numbers (e.g., the amount of money in *Fraud* and grams of the drug in *Drug trafficking*). However, conventional pre-trained

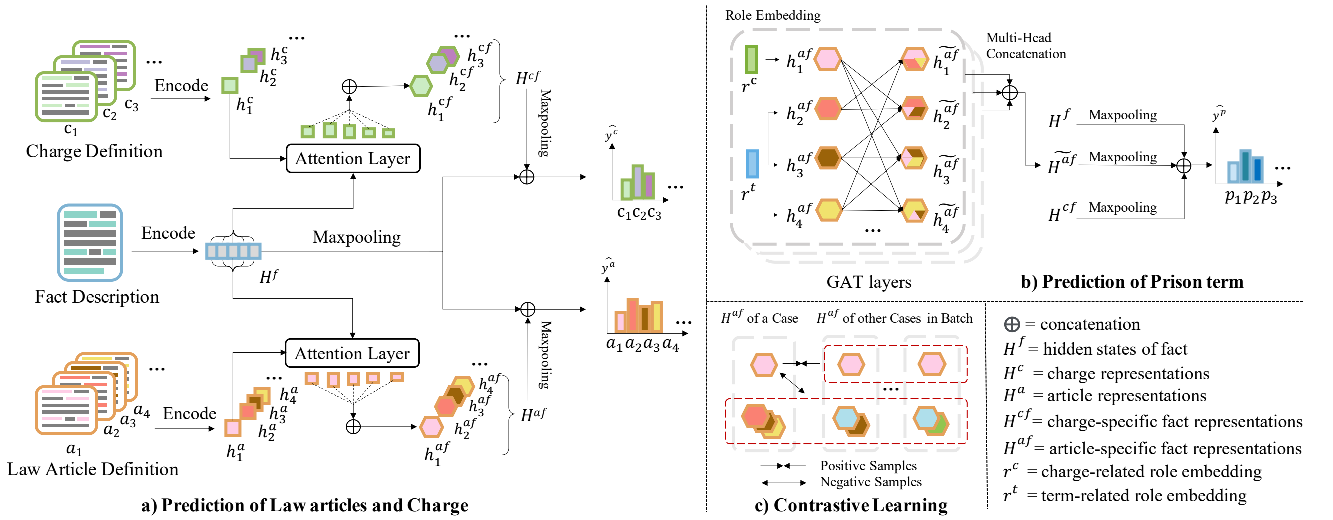

Figure 2: The architecture of ML-LJP. a) is the prediction of law articles and charge, b) is the prediction of prison term, c) shows how to choose the positive samples and negative samples for a certain label for contrastive learning in the training.

language models simply segment the number in the text into sub-words (e.g., segment 43,000 into 430 and ##00), which makes it cumbersome to learn the correct magnitude of numbers [16]. Here we introduce a method to better represent the numbers for LJP.

Since not all numbers are effective (e.g., door number), we first design a unit set $u \in \{gram, yuan, dollar, meter, ...\}$ and locate the effective numbers through the unit set.

Then, we split these located numbers into the mantissa part and exponent part (e.g., split 43,000 into 4.3 and 10^4). Combined with the unit, a number x thus becomes a triplet $(m(x), e(x), u(x))$, where $m(x) \in (-10, 10)$ is the mantissa part, $e(x) \in \{0, 1, 2...9\}$ is the exponent part and $u(x)$ is the unit. For example, "43,000" can be denoted as (4.3, 4, yuan), assuming "yuan" is the following unit.

The $e(x)$ and $u(x)$ can be regarded as normal string tokens and get their embeddings like other tokens. As for the mantissa part, we follow the previous work [16] and represent it as the distance between the mantissa and each element in prototypes $q = \{q_i\}_{i=0}^{d-1} \in \mathbb{R}^d$ where $q_i \in [-10, 10]$. The mantissa embedding $ME(x) \in \mathbb{R}^d$ can be obtained as follow:

$$
\begin{aligned}
q_i &= \frac{10 - (-10)}{d - 1} \times i + (-10), \\
ME(x)_i &= exp\left(-\frac{\|m(x) - q_i\|^2}{\sigma^2}\right).
\end{aligned}
\tag{1}
$$

where σ is a hyperparameter and d is the dimension of the hidden vector. The prototypes in q are uniformly distributed in $[-10, 10]$.

Finally, the embedding of x is the weighted sum of the mantissa embedding, exponent embedding, and unit embedding and we empirically set the weights as 0.2, 0.4, and 0.4 respectively.

The number representation method will not change the architecture of the original encoder and is easy to implement. We apply this method to the following encoders by default.

3.1.2 *Fact Description Encoder.*

Given the fact description of a case in the form of a word sequence $f = \{w_1^f, ..., w_{l_f}^f\}$, we input f into a pre-trained language model and get the hidden vector of fact description as follow:

$$H^f = \text{Encode}(f) \in \mathbb{R}^{l_f \times d}, \tag{2}$$

where $H^f = \{h_1^f, ..., h_{l_f}^f\}$ and d is the dimension.

3.1.3 *Label Definition Encoder.*

Here, we encode the label definitions, which are defined in the Code of Law. Take the law article as an example, to obtain the representations of law article definitions, we encode the word sequences of article definitions $X^a = \{x^{a_1}, ..., x^{a_{n_a}}\}$ through an embedding layer and three transformer layers to obtain the hidden vector for each article:

$$h_i^a = \text{Transformer}(\text{Emb}(x^{a_i})) \in \mathbb{R}^d, \tag{3}$$

where h_i^a is the hidden vector for article a_i. We collect hidden vectors for all of the law articles as $H^a = \{h_1^a, h_2^a, ..., h_{n_a}^a\} \in \mathbb{R}^{n_a \times d}$, where n_a is the number of law articles.

The charge representations $H^c = \{h_1^c, h_2^c, ..., h_{n_c}^c\} \in \mathbb{R}^{n_c \times d}$ are obtained in the same way, where n_c is the number of charges.

3.1.4 *Label-Specific Fact Representation.*

The label-specific fact representation aims to emphasize the label-related information of the fact. Again, taking the law article as an example, in order to emphasize the part of fact relevant to a certain law article, we take the representation of the article H^a as the query, hidden vector sequence H^f as key and value, and do an attention [38] operation to obtain the corresponding article-specific fact representation:

$$H^{af} = \text{Attention}(H^a, H^f, H^f) \in \mathbb{R}^{n_a \times d}, \tag{4}$$

where $H^{af} = \{h_1^{af}, ..., h_{n_a}^{af}\}$ is the hidden vectors for article-specific fact representations and h_i^{af} denotes the part of the fact description that is relevant to article a_i.

Similarly, charge-specific fact representations $H^{cf} = \{h_1^{cf}, ..., h_{n_c}^{cf}\} \in \mathbb{R}^{n_c \times d}$ are obtained the same way as article-specific fact representations.

3.2 Article Prediction

In this section, we use the textual representations obtained from the above calculation for law articles prediction.

3.2.1 ***Article Predictor.*** Given the article-specific fact representations H^{af} and fact representation H^f, we apply max pooling operator over H^{af} and H^f to obtain the pooled article-specific fact representation h^{af} and the context vector h^f of fact description. Then, the concatenation of the h^{af} and h^f is fed into a fully-connected network with sigmoid activation to obtain the predicted results for law article prediction:

$$\begin{aligned} h^{af} &= \text{MaxPooling}(H^{af}) \in \mathbb{R}^d, \\ h^f &= \text{MaxPooling}(H^f) \in \mathbb{R}^d, \\ \hat{y^a} &= \text{sigmoid}(W^a \cdot (h^{af}||h^f)^T + b^a), \end{aligned} \tag{5}$$

where $W^a \in \mathbb{R}^{n_a \times 2d}$ and $b^a \in \mathbb{R}^{n_a}$ are learnable parameters, $||$ denotes concatenation. Then, the predicted law articles are:

$$\hat{a} = \arg\{\hat{y_i^a} > 0.5\}, \quad \hat{y_i^a} \in \hat{y^a}. \tag{6}$$

3.3 Charge Prediction

3.3.1 ***Charge Predictor.*** Similar to the law article prediction, we apply max pooling operator over H^{cf} to obtain the pooled charge-related fact representation h^{cf}. Then, the predicted result for charge prediction is calculated by a fully-connected network but with softmax activation:

$$\begin{aligned} h^{cf} &= \text{MaxPooling}(H^{cf}) \in \mathbb{R}^d, \\ \hat{y^c} &= \text{softmax}(W^c \cdot (h^{cf}||h^f)^T + b^c), \end{aligned} \tag{7}$$

where $W^c \in \mathbb{R}^{n_c \times 2d}$ and $b^c \in \mathbb{R}^{n_c}$ are learnable parameters. Then, the predicted charge is:

$$\hat{c} = \arg\max_i \hat{y_i^c}, \quad \hat{y_i^c} \in \hat{y^c}. \tag{8}$$

3.4 Prison Term Prediction

3.4.1 ***Law Article Role Embedding.*** Since the charge-related and term-related law articles play different roles in the prediction of the prison term, to distinguish them, we add two different role embeddings to every h_i^{af} in H^{af} before the following steps:

$$h_i^{af(r)} = \begin{cases} h_i^{af} + r^c, & \text{if } a_i \in \mathcal{C} \\ h_i^{af} + r^t, & \text{if } a_i \in \mathcal{T}, \end{cases} \tag{9}$$

where $r^c \in \mathbb{R}^d$ and $r^t \in \mathbb{R}^d$ are the learnable role embeddings for charge-related and term-related law articles respectively. $\mathcal{C}$ and $\mathcal{T}$ are the set of charge-related and term-related law articles. Then, we get the role-enhanced article-specific fact representation $H^{af(r)}$.

3.4.2 ***Graph Attention Network.*** To capture the high-order interactions among the multiple law articles in a case, we construct a graph neural network on the law articles. More formally, we define the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, ..., v_n\}$ denotes the nodes which represent the law articles, n denotes the number of nodes in $\mathcal{G}$ and $\mathcal{E}$ denotes the co-occurrence of law articles in a case. For different cases, the $\mathcal{V}$ is the same but the $\mathcal{E}$ is different.

Here, we use the graph attention network (GAT) [39]. GAT uses a multi-head attention mechanism to dynamically integrate neighbor node representations from multi aspects.

In a multi-layer GAT, each GAT layer takes the hidden vectors from previous layer $H^{(l)} = \{h_1^{(l)}, ..., h_n^{(l)}\}$ as inputs and outputs the enhanced representations $H^{(l+1)} = \{h_1^{(l+1)}, ..., h_n^{(l+1)}\}$. In our case, the hidden states $H^{(0)}$ of the first layer are the role-enhanced article-specific fact representations $H^{af(r)}$ and the node number n in the graph is the number of law articles n_a.

To be specific, in one GAT layer, the attention coefficients between the law article a_i and a_j are calculated as:

$$e_{ij} = \sigma(a(Wh_i^{(l)}||Wh_j^{(l)})), \tag{10}$$

$$e'_{ij} = \begin{cases} e_{ij} & (v_i, v_j) \in \mathcal{E} \\ -\infty & otherwise \end{cases} \tag{11}$$

$$\alpha_{ij} = \text{softmax}_j(e'_{ij}) = \frac{exp(e'_{ij})}{\Sigma_{k=1}^n exp(e'_{ik})}, \tag{12}$$

where $W \in \mathbb{R}^{d' \times d}$ and $a \in \mathbb{R}^{2d'}$ are learnable parameters, d' is the output dimension, $\sigma(\cdot)$ denotes LeakyReLU [29] activation and ∞ denotes a very large number. The article pairs that are not co-occurring will be masked since the corresponding coefficients will be close to zero.

Then, the representations from neighbors are aggregated and scaled by the attention coefficients. The outputs of multi-head attention are concatenated:

$$h_i^{(l+1)} = ||_{k=1}^K \sigma\left(\sum_{j=1}^n \alpha_{ij}^k W^k h_j^{(l)}\right) \in \mathbb{R}^{Kd'}, \tag{13}$$

where α_{ij}^k are normalized attention coefficients computed by the k-th head, $W_k \in \mathbb{R}^{d'}$ is the learnable weight matrix and K is the number of heads. Notably, we choose $d' = d/K$ to keep the dimension of hidden vectors same between layers.

After propagation in the GAT, the aggregated article-specific fact representations $\tilde{H^{af}}$ are obtained from the last layer:

$$\tilde{H^{af}} = H^{(L)} = \{h_1^{(L)}, ..., h_n^{(L)}\} \in \mathbb{R}^{n \times d}, \tag{14}$$

where L denotes the number of layers in GAT.

3.4.3 ***Prison Term Predictor.*** Given $\tilde{H^{af}}$, we use max pooling operator to obtain the pooled hidden vector $\tilde{h^{af}}$. Along with the pooled charge-specific fact representation h^{cf} and fact representation h^f from the above calculation, the concatenation of $\tilde{h^{af}}$, h^{cf} and h^f is fed into a fully-connected network with softmax activation to obtain the predicted result for prison term prediction:

$$\begin{aligned} \tilde{h^{af}} &= \text{MaxPooling}(\tilde{H^{af}}) \in \mathbb{R}^d, \\ \hat{y^p} &= \text{softmax}(W^p \cdot (\tilde{h^{af}}||h^{cf}||h^f)^T + b^p), \end{aligned} \tag{15}$$

where $W^p \in \mathbb{R}^{n_p \times 3d}$ and $b^p \in \mathbb{R}^{n_p}$ are learnable parameters. Then, the predicted prison term is:

$$\hat{p} = \arg\max_i \hat{y_i^p}, \quad \hat{y_i^p} \in \hat{y^p}. \tag{16}$$

3.5 Contrastive Learning

In order to empower the ability of our model to distinguish confusing law articles and charges, we follow the previous work [6, 18] to conduct contrastive learning on label representations and label-specific fact representations to learn discriminative features.

Also take the law article as an example, given a mini-batch, we define the set of the ground-truth labels as $Y = \{y_{ij} \in \{0,1\} | i \in \{1,...,N\}, j \in \{1,...,n_a\}\}$, where N is the size of the mini-batch and n_a is the number of law article labels. Then, we select the positive and negative samples for article-specific fact representations whose corresponding $y_{ij} = 1$. Specifically, for the v-th article-specific fact representation of u-th case $h^{af}_{i=u,j=v}$, we define its active label set as $A_{(u,v)} = \{h^{af}_{ij} | y_{ij} = 1\} \backslash h^{af}_{uv}$. Then, the positive and negative sets for h^{af}_{uv} are defined as:

$$\begin{aligned} Pos_{(i=u,j=v)} &= \{h^{af}_{ij} \in A_{(u,v)} | i \in \{1,...,N\}, j = v\}, \\ Neg_{(i=u,j=v)} &= A_{(u,v)} \backslash Pos_{(i=u,j=v)}, \end{aligned} \tag{17}$$

which means we choose article-specific fact representations related to the v-th law article from other samples in $A_{(u,v)}$ as positive samples. The negative set is defined as the remaining representations in $A_{(u,v)}$. Also, we add article representations $H^a = \{h^a_1, h^a_2, ..., h^a_{n_a}\}$ to the positive and negative sets in order to keep article representations discriminative in the same embedding space:

$$\begin{aligned} Pos'_{(i,j)} &= Pos_{(i,j)} \cup \{h^a_j\}, \\ Neg'_{(i,j)} &= Neg_{(i,j)} \cup \{h^a_1, ..., h^a_{j-1}, h^a_{j+1}, ..., h^a_{n_a}\}. \end{aligned} \tag{18}$$

With the above notations, the contrastive loss for the article-specific fact representation h^{af}_{ij} is defined as :

$$\mathcal{L}^{ca}_{ij} = -\sum_{h^+ \in Pos'_{(i,j)}} log \frac{exp(sim(h^{af}_{ij}, h^+)/\tau)}{\Sigma_{h^- \in Neg'_{(i,j)}} exp(sim(h^{af}_{ij}, h^-)/\tau)}, \tag{19}$$

where the similarity function sim is cosine similarity and τ is a temperature hyperparameter. The total loss in the mini-batch is summed over the active label sets:

$$\mathcal{L}^{ca} = \Sigma_{\{i,j|y_{ij}=1\}} \mathcal{L}^{ca}_{ij}. \tag{20}$$

The contrastive loss $\mathcal{L}^{cc}$ for learning discriminative charge-specific fact representations is calculated in the same way.

3.6 Training and Inference

To optimize, we use binary cross-entropy loss for multi-label classification and cross-entropy loss for single-label classification:

$$\begin{aligned} \mathcal{L}^a &= \Sigma^{n_a}_{j=1} [y^a_j log \hat{y}^a_j + (1 - y^a_j) log(1 - \hat{y}^a_j)], \\ \mathcal{L}^c &= \Sigma^{n_c}_{j=1} y^c_j log \hat{y}^c_j, \\ \mathcal{L}^p &= \Sigma^{n_p}_{j=1} y^p_j log \hat{y}^p_j, \end{aligned} \tag{21}$$

where y_j stands for true label and $\hat{y}_j$ is the predicted score for j-th category in each task, n_p is the number of intervals of prison term.

The classification loss and contrastive loss are summed over and weighted by the corresponding weight factor λ as the overall loss:

$$\mathcal{L} = \lambda^a \mathcal{L}^a + \lambda^c \mathcal{L}^c + \lambda^p \mathcal{L}^p + \lambda^{ca} \mathcal{L}^{ca} + \lambda^{cc} \mathcal{L}^{cc}. \tag{22}$$

In GAT module, we use true labels a during training and predicted results $\hat{a}$ during inference for masking.

4 EXPERIMENTS

4.1 Dataset

We conduct our experiments on the dataset of Legal Artificial Intelligence Challenge (LAIC) 2021. For one sample, it contains fact description, court view, charge and prison term. We design rules to extract relevant articles from the text of the court view. For data processing, we first filter out samples with fewer than 10 meaningful words in fact description. Then, we only keep the law articles and charges that apply to not less than 50 case samples. Consistency with previous work [56, 60], prison terms are divided into non-overlapping intervals. We randomly divide the processed data into training set, validation set and test set according to the ratio of 8: 1: 1. The statistics of the LAIC2021 dataset are shown in Tab. 2.

Type	Result
# Cases	98962
# Law Articles	70
# Charges	42
# Prison Term	9
Avg. # Tokens in Fact Description	473.1
Avg. # Tokens in Article Definition	157.5
Avg. # Tokens in Charge Definition	112.0
Avg. # Law Articles in a Case	3.1

Table 2: Statistics of dataset.

4.2 Baselines

We implement the following baselines for comparison:

CNN [19] extracts text features through convolutional operations with different kernels for text classification; **HARNN** [54] is a RNN-based model with two levels of attention mechanisms for aggregating words to sentences and sentences to documents; **LSAN** [50] uses document and label to learn the label-specific document representation with the aid of self-attention and label-attention mechanism for multi-label text classification. **BERT** [7] and **Electra** [5] [4] are language models pre-trained on large corpus and can be fine-tuned on downstream tasks.

We also take the following LJP methods as baselines:

TopJudge [60] is a topological multitask learning model that captures dependencies between subtasks in LJP; **CPTP** [3] filters and aggregates charge-specified information with a gating mechanism to enhance the performance of predicting the prison term. **FLA** [26] uses an attention-based neural network to model the charge prediction and the article extraction with a legal basis; **LADAN** [51] uses graph distillation to extract discriminative features for distinguishing confusing charges and law articles; **NeurJudge** [56] uses the results of intermediate subtasks to divide the factual description into different parts for making predictions for other subtasks; **R-Former** [8] formalizes LJP as a node classification problem over a global consistency graph and relational learning are introduced to solve the node classification problem. **EPM** [10] leverages annotated event extraction data to locate the key event and introduces constraints among the subtasks; **DPAM** [43] considers multiple charge-related law articles and incorporates a pair-wise attention module for alleviating the label imbalance problem; **HMN** [42]

[4] Can be downloaded in https://huggingface.co/hfl/chinese-legal-electra-base-discriminator.

Methods	Micro				Macro			
	P	R	F1	Jaccard	P	R	F1	Jaccard
CNN [19]	82.45	83.50	82.97	70.90	71.47	56.21	60.41	53.49
BERT [7]	<u>85.43</u>	82.31	83.84	<u>74.24</u>	71.91	64.36	66.49	60.70
Electra [5]	84.57	<u>84.19</u>	<u>84.38</u>	72.98	71.97	65.41	67.09	61.17
HAN [54]	84.52	81.44	82.96	70.88	71.26	61.38	64.77	55.56
FLA [26]	83.74	82.87	83.30	71.39	68.33	64.47	65.38	56.13
LSAN [50]	85.26	82.69	83.96	72.35	72.60	61.30	63.93	57.02
DPAM [43]	85.17	80.95	83.95	72.34	64.81	66.70	65.67	<u>63.98</u>
HMN [42]	85.02	83.03	84.01	72.43	<u>72.62</u>	<u>67.34</u>	<u>68.88</u>	61.97
ML-LJP	**85.71**	**85.71**	**85.71**	**75.00**	**75.71**	**71.71**	**73.21**	**65.41**
ML-LJP w/o Def	84.20	83.19	83.75	73.54	71.82	67.23	68.11	60.43
ML-LJP w/o CL	84.55	84.19	84.30	73.72	72.12	68.84	69.47	61.63
ML-LJP w/o GAT	85.21	84.76	84.92	73.90	72.04	71.35	70.70	63.01
ML-LJP w/o Role	84.69	85.31	84.82	74.54	73.59	71.00	71.73	62.58

Table 3: Results of law article prediction, <u>underline</u> denotes the second best.

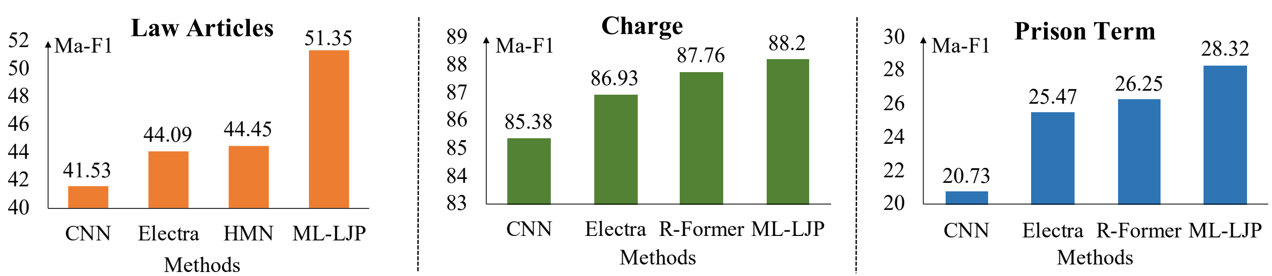

Figure 3: Evaluation on low-frequency cases.

predicts multiple charge-related law articles and decomposes the semantics of articles into different components.

We do ablation experiments as follows:

ML-LJP w/o Def denotes that we remove the charge and law article definitions from the input, and replace the label definition encoder with a randomly initialized label embedding layer; **ML-LJP w/o CL** denotes that the contrastive loss is removed; **ML-LJP w/o GAT** denotes that the GAT module is removed; **ML-LJP w/o Role** denotes that the role embeddings of the law articles are removed.

4.3 Experiment Settings

In the experiments, we set the maximum document length of fact description to 512, law article definition and charge definition to 200. We choose Adam[20] as the optimizer for all the models. To ensure reproducibility, all the models are set with seed=2022.

For DPAM, HMN, LADAN, NeurJudge, FLA, EPM, LSAN, and R-Former, we use the training setup from the original paper.

For the baselines without a pre-trained language model, we use word embeddings pre-trained on CAIL2018 dataset with the embedding size as 300, the learning rate is initialized as 10^{-4} for Adam optimizer. For the baselines with a pre-trained language model, the learning rate is initialized as 10^{-5} for Adam optimizer.

We use Electra-base as the encoder in our model. We set the batch size to 64 and initialize the learning rate of Adam optimizer to 10^{-5} for the pre-trained parameters and 10^{-4} for the non-pre-trained parameters. We set the number of layers L in GAT to 2, the hyper-parameter σ for mantissa embedding to 0.5 and the temperature hyperparameter τ to 0.07.

For the hyperparameters, λ, in the loss function, the best setting is {1.0, 1.0, 1.0, 0.5, 0.5} for $\{\lambda^a, \lambda^c, \lambda^p, \lambda^{ca}, \lambda^{cc}\}$. All the experiments have been performed on a server with 2×3090 GPU.

For the metrics, we employ Precision (P), Recall (R), F1 score (F1), and Jaccard under micro and macro for law articles prediction and accuracy (Acc), macro Precision (MaP), macro Recall (MaR), and macro F1 score (MaF) for charge and prison term prediction.

4.4 Experiments Results

We analyze the experimental results in this section.

4.4.1 ***Comparison against baselines.*** From Tab. 3 and Tab. 4, we have the following observations: 1) For law article prediction, we compare ML-LJP against multi-label classification baselines (e.g., DPAM and HMN), and ML-LJP achieves the best performance (e.g., outperforms the HMN by 2.02% in micro F1 and 6.28% in macro F1), which shows the advantage of incorporating law article definitions to extract the article-specific fact representations. 2) For the charge prediction task, most of the models achieve a high score. Despite this, ML-LJP still has a competitive performance on charge prediction. 3) On the task of prison term prediction, our method outperforms other baselines to a large extent. In particular, ML-LJP achieves a 10.07% relative improvement over the best baseline

Methods	Charge				Prison Term			
	Acc	MaP	MaR	MaF	Acc	MaP	MaR	MaF
CNN [19]	96.41	91.91	90.17	91.03	34.14	33.19	32.69	32.94
BERT [7]	96.72	93.25	88.82	90.98	41.83	41.88	41.43	41.65
Electra [5]	96.29	<u>94.30</u>	91.56	92.91	42.56	42.29	41.39	41.84
HARNN [54]	95.11	88.31	84.93	86.59	33.43	32.07	30.44	31.23
TopJudge [60]	96.46	91.97	90.17	91.06	38.97	38.90	35.16	36.94
CPTP [3]	-	-	-	-	35.46	35.67	33.11	34.34
LADAN [51]	96.21	91.20	91.51	91.35	41.28	40.55	38.73	39.62
NeurJudge [56]	94.21	88.43	84.73	86.54	37.12	38.16	34.07	36.00
EPM [10]	97.11	93.89	92.63	93.11	40.48	38.99	38.34	38.38
R-Former [8]	<u>97.49</u>	93.96	**94.13**	<u>94.04</u>	<u>43.62</u>	<u>43.14</u>	<u>43.21</u>	<u>43.17</u>
ML-LJP	**97.54**	**95.56**	<u>93.73</u>	**94.64**	**47.63**	**48.39**	**46.68**	**47.52**
ML-LJP w/o Def	96.86	92.41	91.94	92.17	46.30	45.18	43.56	44.35
ML-LJP w/o CL	97.11	93.71	92.80	93.25	45.99	45.38	44.76	45.06
ML-LJP w/o GAT	97.19	93.95	93.06	93.50	44.44	43.97	42.87	43.41
ML-LJP w/o Role	97.14	93.41	92.85	92.97	44.81	44.56	42.46	43.48

Table 4: Results of charge and prison term prediction, <u>underline</u> denotes the second best.

R-former in macro F1-score, which proves the effectiveness of taking both charge-related law articles and term-related law articles into consideration and learning the interactions among them. 4) Compared to the single-task model (e.g, CPTP), models with multi-task learning show their superiority in the LJP task. 5) Overall, the ML-LJP achieves the best performance in all the tasks of LJP.

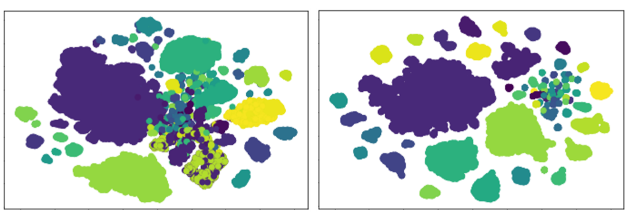

Figure 4: t-SNE visualization of article-specific fact representations without and with contrastive learning.

4.4.2 ***Ablation study.*** In the ablation study, we analyze the impact of each module used in ML-LJP. From Tab. 3 and Tab. 4, we can conclude that: 1) The performance gap between ML-LJP and ML-LJP w/o Def exhibits the effectiveness of using the label definitions. For example, the Ma-P of charge prediction drops from 95.56% to 92.41%. 2) The performance gap between ML-LJP and ML-LJP w/o CL illustrates that the model can be confused by some similar law articles or charges and leads to a drop in performance. For example, without CL, the Ma-F1 of law article prediction drops from 73.21% to 69.47%. 3) The performance gap between ML-LJP and ML-LJP w/o GAT shows the rationality of learning the interactions among the multiple law articles. 4) Without the role embedding, ML-LJP w/o Role drops the performance, which proves the necessity of distinguishing the two types of law articles.

Fig. 4 shows the visualization of the corresponding article-specific fact representations of cases from the test set, which demonstrates that labels are more discriminative with contrastive learning.

4.4.3 ***Low Frequency Scenarios.*** Moreover, we evaluate the performance of low frequency cases. Here, we take the cases with the lowest 25% frequency charge as the low frequency cases. As shown in Fig. 3, ML-LJP outperforms other baselines by a large margin in all the tasks, especially in the perdition of the law articles and the prison term (e.g., compare to the best baseline HMN, the Ma-F1 of law articles improve from 44.45% to 51.35%). The performance improvement of ML-LJP on low frequency cases shows ML-LJP can mitigate the impact of data imbalance and makes the model robust.

Methods	ACC	MaP	MaR	MaF
Replace	43.63	42.48	41.31	41.89
Subword	46.45	46.58	44.35	45.43
w/o Unit	46.81	46.50	45.62	46.08
ML-LJP	**47.63**	**48.39**	**46.68**	**47.52**

Table 5: Analysis on Number Representation.

4.4.4 ***Analysis on Number Representation.*** Since numbers are important to legal cases, we conduct experiments to evaluate the effectiveness of the number representation method (in Sec 3.1.1) through the task of prison term prediction. Specifically, we compare it with two other settings:

Replace means that we replace all the meaningful numbers in legal text with a "*number*" token. For example, the number 43,000 is represented as [number]; **Subword** means that the numbers are segmented into subwords by the tokenizer as conventional pre-trained language models do. For example, the number 43,000 is represented as [430, ##00]; **w/o Unit** means that the unit embedding is removed, the weights of mantissa embedding and exponent embedding are set as 0.3 and 0.7.

As shown in Tab. 5, the poor performance of **Replace** demonstrates that the meaningful numbers in legal text are essential, and compared to the conventional method (e.g. **Subword**), our number representation method can capture the meaning of the number more precisely. Also, the performance drop in **w/o Unit** shows that distinguishing numbers in different unit is beneficial.

Fact Description	It was found that on October 13, 2013 at around 11:00 a.m., the defendant A posed as a relative of B (who had been sentenced to imprisonment) and used the excuse that B's condition had worsened... He cheated the victim C of RMB 36,600 yuan and hospital fees of RMB 7,000 yuan. After B arrived at the case to withdraw the stolen money RMB 5000 yuan... The defendant A then rushed to XXX Hospital posing as B's fellow countryman, and in the process of cheating the doctor at the service station, i.e. victim D, of his belongings, A and B were caught on the spot because D called the police... The defendant A failed to obtain property in the fraud on victim D, which was an attempted crime. It was also found that Defendant A was sentenced to one year and eight months in prison by XXX People's Court on December 14, 2011 for the crime of extortion. He is a repeat offender. The defendant A confessed his crime truthfully after his return to the court.			
Judgment		Ground Truth	R-former	ML-LJP
	Law Articles	Article 266, 67, 65, 23	Article 266	Article 266, 67, 65, 23
	Charge	Fraud	Fraud	Fraud
	Prison Term	21 months	24 ~ 36 months ❌	12 ~ 24 months ✅

Figure 5: Case Study.

4.5 Case Study

We visualize the attention of ML-LJP's attention module. As Fig. 5 shows, in a legal case, the state-of-the-art baseline R-former considers only the charge-related law article (e.g., *Article 266*) and predicts the wrong prison term. Benefiting from the multiple law article prediction, ML-LJP predicts the correct prison term. Moreover, we visualize the attention weights on the fact description when calculating the corresponding article-specific fact representation. We find that different law articles focus on different parts. For example, *Article 266* describes the crime of fraud, and the words like *fraud* and *cheat* are assigned more attention weights, which proves the effectiveness of the article-specific fact representation.

4.6 Error Analysis

To explore the limitations of our model, we conduct an error analysis. After analyzing the randomly selected 200 samples with wrong judgment predictions, we make the following observations: 1) The cases in the dataset do not contain all the information of real legal cases. The lack of some information will have an impact on the prediction of judgments. 2) In different regions, the judgment on charges and the reference to law articles are mostly consistent, but when it comes to the determination of prison terms, there may be a difference due to some local principles. The missing of these local principles can limit the performance of the model. 3) In addition, every year, the Supreme People's Court issues some amendments to the law, so the same criminal fact may have different prison terms in different years, which can be another reason for hindering the model's performance.

To address these problems, constructing a legal database and injecting more meta legal knowledge into the model in an appropriate way can be promising in the future.

5 ETHICAL ISSUE DISCUSSION

In this section, we make an ethical discussion the clarify the purpose of our work. With the development of LegalAI, ethical issues become more important since any subtle miscalculation may trigger serious consequences [46]. Therefore, the ethical concerns should be further investigated.

Firstly, the target user of LJP is the trial judge, who suffers from a 'daunting workload' (e.g., the trial judge has to close around 250 cases a year [9]). In such circumstances, the proposed algorithm aims to offer suggestions to the judges but should **never replace** the human judges. Indeed, our purpose is to provide assistance to the judges and improve their work efficiency. In practical use, human judges should be the final safeguard to protect justice fairness.

In addition, since the models are trained with a large dataset, with the development of AI techniques, LJP models have the potential to protect the principle of "treating like cases alike" [56] in the future.

In this paper, we only make an algorithmic investigation, and there still exist some risks (e.g., lack of interactivity and interpretability). Therefore, the model will not be used in real legal scenarios by far.

6 CONCLUSION AND FUTURE WORK

In this paper, we explore the legal judgment prediction (LJP) problem and take both charge-related law articles and term-related articles into consideration. Following the real legal process, we expand the law article prediction task and propose a novel multi-law-based LJP (ML-LJP) method. The ML-LJP makes use of the label definition and extracts the label-specific features of the fact. To better distinguish the similar labels, we conduct contrastive learning in the training. Then, a graph attention network (GAT) is applied to capture the high-order interactions among multiple law articles. Moreover, we propose a simple yet effective number representation method to locate and better encode the numbers in the fact. A series of experiments show the effectiveness of our method over other baselines. Finally, we make an ethical discussion of our work due to the sensitivity of LegalAI.

In the future, we can explore the following directions: 1) Based on the number representation method, we can try to use numerical reasoning to better utilize the numbers in the fact. 2) Add external knowledge (e.g., a logic graph) to better learn the interactions among the multiple law articles.

7 ACKNOWLEDGEMENTS

This work was supported in part by National Key Research and Development Program of China (2022YFC3340900), National Natural Science Foundation of China (U20A20387, 62006207, 62037001), Young Elite Scientists Sponsorship Program by CAST (2021QNRC001), the StarryNight Science Fund of Zhejiang University Shanghai Institute for Advanced Study (SN-ZJU-SIAS-0010).

REFERENCES

[1] Philip Bachman, R. Devon Hjelm, and William Buchwalter. 2019. Learning Representations by Maximizing Mutual Information Across Views. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 15509–15519. https://proceedings.neurips.cc/paper/2019/hash/ddf354219aac374f1d40b7e760ee5bb7-Abstract.html

[2] Paheli Bhattacharya, Kripabandhu Ghosh, Arindam Pal, and Saptarshi Ghosh. 2020. Hier-SPCNet: A Legal Statute Hierarchy-based Heterogeneous Network for Computing Legal Case Document Similarity. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, Jimmy X. Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (Eds.). ACM, 1657–1660. https://doi.org/10.1145/3397271.3401191

[3] Huajie Chen, Deng Cai, Wei Dai, Zehui Dai, and Yadong Ding. 2019. Charge-Based Prison Term Prediction with Deep Gating Network. *empirical methods in natural language processing* (2019).

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event (Proceedings of Machine Learning Research, Vol. 119)*. PMLR, 1597–1607. http://proceedings.mlr.press/v119/chen20j.html

[5] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. https://openreview.net/forum?id=r1xMH1BtvB

[6] Son Duy Dao, Ethan Zhao, D.Q. Phung, and Jianfei Cai. 2021. Multi-Label Image Classification with Contrastive Learning. *ArXiv* abs/2107.11626 (2021).

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, 4171–4186. https://doi.org/10.18653/v1/n19-1423

[8] Qian Dong and Shuzi Niu. 2021. Legal Judgment Prediction via Relational Learning. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (Eds.). ACM, 983–992. https://doi.org/10.1145/3404835.3462931

[9] Xinyu Duan, Yating Zhang, Lin Yuan, Xin Zhou, Xiaozhong Liu, Tianyi Wang, Ruocheng Wang, Qiong Zhang, Changlong Sun, and Fei Wu. 2019. Legal Summarization for Multi-role Debate Dialogue via Controversy Focus Mining and Multi-task Learning. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, Wenwu Zhu, Dacheng Tao, Xueqi Cheng, Peng Cui, Elke A. Rundensteiner, David Carmel, Qi He, and Jeffrey Xu Yu (Eds.). ACM, 1361–1370. https://doi.org/10.1145/3357384.3357940

[10] Yi Feng, Chuanyi Li, and Vincent Ng. 2022. Legal Judgment Prediction via Event Extraction with Constraints. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, 648–664. https://doi.org/10.18653/v1/2022.acl-long.48

[11] Leilei Gan, Kun Kuang, Yi Yang, and Fei Wu. 2021. Judgment Prediction via Injecting Legal Knowledge into Neural Networks. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 12866–12874. https://ojs.aaai.org/index.php/AAAI/article/view/17522

[12] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (Eds.). Association for Computational Linguistics, 6894–6910. https://doi.org/10.18653/v1/2021.emnlp-main.552

[13] Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting Numerical Reasoning Skills into Language Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (Eds.). Association for Computational Linguistics, 946–958. https://doi.org/10.18653/v1/2020.acl-main.89

[14] William L. Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 1024–1034. https://proceedings.neurips.cc/paper/2017/hash/5dd9db5e033da9c6fb5ba83c7a7ebea9-Abstract.html

[15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. Computer Vision Foundation / IEEE, 9726–9735. https://doi.org/10.1109/CVPR42600.2020.00975

[16] Zhihua Jin, Xin Jiang, Xingbo Wang, Qun Liu, Yong Wang, Xiaozhe Ren, and Huamin Qu. 2021. NumGPT: Improving Numeracy Ability of Generative Pre-trained Models. *arXiv: Computation and Language* (2021).

[17] Shen Kai, Lingfei Wu, Siliang Tang, Yueting Zhuang, Zhuoye Ding, Yun Xiao, Bo Long, et al. 2021. Learning to generate visual questions with noisy supervision. *Advances in Neural Information Processing Systems* 34 (2021), 11604–11617.

[18] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised Contrastive Learning. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.). https://proceedings.neurips.cc/paper/2020/hash/d89a66c7c80a29b1bdbab0f2a1a94af8-Abstract.html

[19] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, Alessandro Moschitti, Bo Pang, and Walter Daelemans (Eds.). ACL, 1746–1751. https://doi.org/10.3115/v1/d14-1181

[20] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1412.6980

[21] Fred Kort. 1957. Predicting Supreme Court decisions mathematically: A quantitative analysis of the "right to counsel" cases. *American Political Science Review* 51, 1 (1957), 1–12.

[22] Mengze Li, Tianbao Wang, Haoyu Zhang, Shengyu Zhang, Zhou Zhao, Jiaxu Miao, Wenqiao Zhang, Wenming Tan, Jin Wang, Peng Wang, et al. 2022. End-to-End Modeling via Information Tree for One-Shot Natural Language Spatial Video Grounding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 8707–8717.

[23] Yuxiao Lin, Yuxian Meng, Xiaofei Sun, Qinghong Han, Kun Kuang, Jiwei Li, and Fei Wu. 2021. BertGCN: Transductive Text Classification by Combining GCN and BERT. *CoRR* abs/2105.05727 (2021). arXiv:2105.05727 https://arxiv.org/abs/2105.05727

[24] Dugang Liu, Weihao Du, Lei Li, Weike Pan, and Zhong Ming. 2022. Augmenting Legal Judgment Prediction with Contrastive Case Relations. In *Proceedings of the 29th International Conference on Computational Linguistics, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, Nicoletta Calzolari, Chu-Ren Huang, Hansaem Kim, James Pustejovsky, Leo Wanner, Key-Sun Choi, Pum-Mo Ryu, Hsin-Hsi Chen, Lucia Donatelli, Heng Ji, Sadao Kurohashi, Patrizia Paggio, Nianwen Xue, Seokhwan Kim, Younggyun Hahm, Zhong He, Tony Kyungil Lee, Enrico Santus, Francis Bond, and Seung-Hoon Na (Eds.). International Committee on Computational Linguistics, 2658–2667. https://aclanthology.org/2022.coling-1.235

[25] Jueqing Lu, Lan Du, Ming Liu, and Joanna Dipnall. 2020. Multi-label Few/Zero-shot Learning with Knowledge Aggregated from Multiple Label Graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, 2935–2943. https://doi.org/10.18653/v1/2020.emnlp-main.235

[26] Bingfeng Luo, Yansong Feng, Jianbo Xu, Xiang Zhang, and Dongyan Zhao. 2017. Learning to Predict Charges for Criminal Cases with Legal Basis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, Martha Palmer, Rebecca Hwa, and Sebastian Riedel (Eds.). Association for Computational Linguistics, 2727–2736. https://doi.org/10.18653/v1/d17-1289

[27] Qianwen Ma, Chunyuan Yuan, Wei Zhou, and Songlin Hu. 2021. Label-Specific Dual Graph Neural Network for Multi-Label Text Classification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, 3855–3864. https://doi.org/10.18653/v1/2021.acl-long.298

[28] Xinyin Ma, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Weiming Lu. 2021. MuVER: Improving First-Stage Entity Retrieval with Multi-View Entity Representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2617–2624. https://doi.org/10.18653/v1/2021.emnlp-main.205

[29] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. 2013. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, Vol. 30. Citeseer, 3.

[30] Mihai Masala, Radu Cristian Alexandru Iacob, Ana Sabina Uban, Marina Cidota, Horia Velicu, Traian Rebedea, and Marius Popescu. 2021. jurBERT: A Romanian BERT Model for Legal Judgement Prediction. In *Proceedings of the Natural Legal Language Processing Workshop 2021*. Association for Computational Linguistics, Punta Cana, Dominican Republic, 86–94. https://doi.org/10.18653/v1/2021.nllp-1.8

[31] Joel Niklaus, Ilias Chalkidis, and Matthias Stürmer. 2021. Swiss-Judgment-Prediction: A Multilingual Legal Judgment Prediction Benchmark. *CoRR* abs/2110.00806 (2021). arXiv:2110.00806 https://arxiv.org/abs/2110.00806

[32] Christos Papaloukas, Ilias Chalkidis, Konstantinos Athinaios, Despina Pantazi, and Manolis Koubarakis. 2021. Multi-granular Legal Topic Classification on Greek Legislation. In *Proceedings of the Natural Legal Language Processing Workshop 2021*. Association for Computational Linguistics, Punta Cana, Dominican Republic, 63–75. https://doi.org/10.18653/v1/2021.nllp-1.6

[33] Jeffrey A Segal. 1984. Predicting Supreme Court cases probabilistically: The search and seizure cases, 1962-1981. *American Political Science Review* 78, 4 (1984), 891–900.

[34] Yongliang Shen, Xinyin Ma, Zeqi Tan, Shuai Zhang, Wen Wang, and Weiming Lu. 2021. Locate and Label: A Two-stage Identifier for Nested Named Entity Recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Online, 2782–2794. https://doi.org/10.18653/v1/2021.acl-long.216

[35] Dhanasekar Sundararaman, Shijing Si, Vivek Subramanian, Guoyin Wang, Devamanyu Hazarika, and Lawrence Carin. 2020. Methods for Numeracy-Preserving Word Embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, 4742–4753. https://doi.org/10.18653/v1/2020.emnlp-main.384

[36] Avijit Thawani, Jay Pujara, Filip Ilievski, and Pedro A. Szekely. 2021. Representing Numbers in NLP: a Survey and a Vision. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (Eds.). Association for Computational Linguistics, 644–656. https://doi.org/10.18653/v1/2021.naacl-main.53

[37] S Sidney Ulmer. 1963. Quantitative analysis of judicial processes: Some practical and theoretical applications. *Law and Contemporary Problems* 28, 1 (1963), 164–184.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 5998–6008. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[39] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. https://openreview.net/forum?id=rJXMpikCZ

[40] Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do NLP models know numbers? probing numeracy in embeddings. *arXiv preprint arXiv:1909.07940* (2019).

[41] Haifeng Wang, Jiwei Li, Hua Wu, Eduard Hovy, and Yu Sun. 2022. Pre-Trained Language Models and Their Applications. *Engineering* (2022).

[42] Pengfei Wang, Yu Fan, Shuzi Niu, Ze Yang, Yongfeng Zhang, and Jiafeng Guo. 2019. Hierarchical Matching Network for Crime Classification. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, Benjamin Piwowarski, Max Chevalier, Éric Gaussier, Yoelle Maarek, Jian-Yun Nie, and Falk Scholer (Eds.). ACM, 325–334. https://doi.org/10.1145/3331184.3331223

[43] Pengfei Wang, Ze Yang, Shuzi Niu, Yongfeng Zhang, Lei Zhang, and Shaozhang Niu. 2018. Modeling Dynamic Pairwise Attention for Crime Classification over Legal Articles. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, Kevyn Collins-Thompson, Qiaozhu Mei, Brian D. Davison, Yiqun Liu, and Emine Yilmaz (Eds.). ACM, 485–494. https://doi.org/10.1145/3209978.3210057

[44] Yitong Wang, Fei Cai, Zhiqiang Pan, and Chengyu Song. 2023. Self-supervised graph learning with target-adaptive masking for session-based recommendation. *Frontiers of Information Technology & Electronic Engineering* 24, 1 (2023), 73–87.

[45] Bin Wei, Kun Kuang, Changlong Sun, Jun Feng, Yating Zhang, Xinli Zhu, Jianghong Zhou, Yinsheng Zhai, and Fei Wu. 2022. A full-process intelligent trial system for smart court. *Frontiers of Information Technology & Electronic Engineering* 23, 2 (2022), 186–206.

[46] Yiquan Wu, Kun Kuang, Yating Zhang, Xiaozhong Liu, Changlong Sun, Jun Xiao, Yueting Zhuang, Luo Si, and Fei Wu. 2020. De-Biased Court's View Generation with Causality. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, 763–780. https://doi.org/10.18653/v1/2020.emnlp-main.56

[47] Yiquan Wu, Yifei Liu, Weiming Lu, Yating Zhang, Jun Feng, Changlong Sun, Fei Wu, and Kun Kuang. 2022. Towards Interactivity and Interpretability: A Rationale-based Legal Judgment Prediction Framework. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 4787–4799. https://aclanthology.org/2022.emnlp-main.316

[48] Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Yansong Feng, Xianpei Han, Zhen Hu, Heng Wang, and Jianfeng Xu. 2018. CAIL2018: A Large-Scale Legal Dataset for Judgment Prediction. *CoRR* abs/1807.02478 (2018). arXiv:1807.02478 http://arxiv.org/abs/1807.02478

[49] Lin Xiao, Xin Huang, Boli Chen, and Liping Jing. 2019. Label-Specific Document Representation for Multi-Label Text Classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, 466–475. https://doi.org/10.18653/v1/D19-1044

[50] Lin Xiao, Xin Huang, Boli Chen, and Liping Jing. 2019. Label-Specific Document Representation for Multi-Label Text Classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, 466–475. https://doi.org/10.18653/v1/D19-1044

[51] Nuo Xu, Pinghui Wang, Long Chen, Li Pan, Xiaoyan Wang, and Junzhou Zhao. 2020. Distinguish Confusing Law Articles for Legal Judgment Prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (Eds.). Association for Computational Linguistics, 3086–3095. https://doi.org/10.18653/v1/2020.acl-main.280

[52] Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. ConSERT: A Contrastive Framework for Self-Supervised Sentence Representation Transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, 5065–5075. https://doi.org/10.18653/v1/2021.acl-long.393

[53] Yi Yang, Yueting Zhuang, and Yunhe Pan. 2021. Multiple knowledge representation for big data artificial intelligence: framework, applications, and case studies. *Frontiers of Information Technology & Electronic Engineering* 22, 12 (2021), 1551–1558.

[54] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical Attention Networks for Document Classification. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, Kevin Knight, Ani Nenkova, and Owen Rambow (Eds.). The Association for Computational Linguistics, 1480–1489. https://doi.org/10.18653/v1/n16-1174

[55] Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph Convolutional Networks for Text Classification. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 7370–7377. https://doi.org/10.1609/aaai.v33i01.33017370

[56] 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 *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (Eds.). ACM, 973–982. https://doi.org/10.1145/3404835.3462826

[57] Han Zhang, Zhicheng Dou, Yutao Zhu, and Ji-Rong Wen. 2023. Contrastive Learning for Legal Judgment Prediction. *ACM Trans. Inf. Syst.* (jan 2023). https://doi.org/10.1145/3580489 Just Accepted.

[58] Xikun Zhang, Deepak Ramachandran, Ian Tenney, Yanai Elazar, and Dan Roth. 2020. Do Language Embeddings capture Scales?. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2020, Online, November 2020*, Afra Alishahi, Yonatan Belinkov, Grzegorz Chrupala, Dieuwke Hupkes, Yuval Pinter, and Hassan Sajjad

(Eds.). Association for Computational Linguistics, 292–299. https://doi.org/10.18653/v1/2020.blackboxnlp-1.27

[59] Yu Zhang, Zhihong Shen, Chieh-Han Wu, Boya Xie, Junheng Hao, Ye-Yi Wang, Kuansan Wang, and Jiawei Han. 2022. Metadata-Induced Contrastive Learning for Zero-Shot Multi-Label Text Classification. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, Frédérique Laforest, Raphaël Troncy, Elena Simperl, Deepak Agarwal, Aristides Gionis, Ivan Herman, and Lionel Médini (Eds.). ACM, 3162–3173. https://doi.org/10.1145/3485447.3512174

[60] 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, Brussels, Belgium, October 31 - November 4, 2018*, Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (Eds.). Association for Computational Linguistics, 3540–3549. https://doi.org/10.18653/v1/d18-1390

[61] Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. JEC-QA: A Legal-Domain Question Answering Dataset. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020.* AAAI Press, 9701–9708. https://ojs.aaai.org/index.php/AAAI/article/view/6519

[62] Ming Zhou, Nan Duan, Shujie Liu, and Heung-Yeung Shum. 2020. Progress in neural NLP: modeling, learning, and reasoning. *Engineering* 6, 3 (2020), 275–290.

[63] Siying Zhou, Yifei Liu, Yiquan Wu, Kun Kuang, Chunyan Zheng, and Fei Wu. 2022. Similar Case Based Prison Term Prediction. In *Artificial Intelligence*, Lu Fang, Daniel Povey, Guangtao Zhai, Tao Mei, and Ruiping Wang (Eds.). Springer Nature Switzerland, Cham, 284–297.

[64] Yueting Zhuang, Ming Cai, Xuelong Li, Xiangang Luo, Qiang Yang, and Fei Wu. 2020. The next breakthroughs of artificial intelligence: The interdisciplinary nature of AI. *Engineering* 6, 3 (2020), 245–247.