

# Enhancing Court View Generation with Knowledge Injection and Guidance

Ang Li<sup>1</sup>, Yiquan Wu<sup>1</sup>, Yifei Liu<sup>1</sup>, Fei Wu<sup>1,2</sup>, Ming Cai<sup>1†</sup>, Kun Kuang<sup>1†</sup>

<sup>1</sup> College of Computer Science and Technology, Zhejiang University

<sup>2</sup> Shanghai Institute for Advanced Study of Zhejiang University  
{liangrex, wuyiquan, liuyifei, wufei, cm, kunkuang}@zju.edu.cn

## Abstract

Court View Generation (CVG) is a challenging task in the field of Legal Artificial Intelligence (LegalAI), which aims to generate court views based on the plaintiff claims and the fact descriptions. While Pretrained Language Models (PLMs) have showcased their prowess in natural language generation, their application to the complex, knowledge-intensive domain of CVG often reveals inherent limitations. In this paper, we present a novel approach, named Knowledge Injection and Guidance (KIG), designed to bolster CVG using PLMs. To efficiently incorporate domain knowledge during the training stage, we introduce a knowledge-injected prompt encoder for prompt tuning, thereby reducing computational overhead. Moreover, to further enhance the model's ability to utilize domain knowledge, we employ a generating navigator, which dynamically guides the text generation process in the inference stage without altering the model's architecture, making it readily transferable. Comprehensive experiments on real-world data demonstrate the effectiveness of our approach compared to several established baselines, especially in the responsiveness of claims, where it outperforms the best baseline by 11.87%.

**Keywords:** Natural Language Generation, Knowledge Discovery, Court View Generation

## 1. Introduction

In recent years, artificial intelligence has been applied in the legal domain. Legal artificial intelligence (LegalAI) focuses on applying methods of artificial intelligence to benefit legal tasks (Zhong et al., 2018, 2019; Yue et al., 2021b; Feng et al., 2022; Liu et al., 2023; Wu et al., 2023). Courts' views can be regarded as interpretations of case judgments, providing specific legal rules applicable to individual cases. Courts' views help ensure fairness and justice in judgments and Court View Generation (CVG) is considered one of the most critical functions in LegalAI.

CVG is a distinctive natural language generation (NLG) task, which aims to generate court views based on the plaintiff claims and the fact descriptions. Existing NLG methods rely on the extensive parameters of pretrained language models (PLMs) to achieve impressive generative capabilities. These methods can be employed to address the CVG task, but their effectiveness may be compromised by insufficient domain-specific knowledge. For example, Fig. 1 shows the court views summarized by the judge and generated by BART (Lewis et al., 2019) in a real case. The judge provides a court view that accurately responds to the plaintiff claim based on their claim-related knowledge. However, BART might mistakenly interpret "interest" as "curiosity" due to its lack of understanding of claim-related knowledge, thereby failing to recognize the word's importance, resulting in the

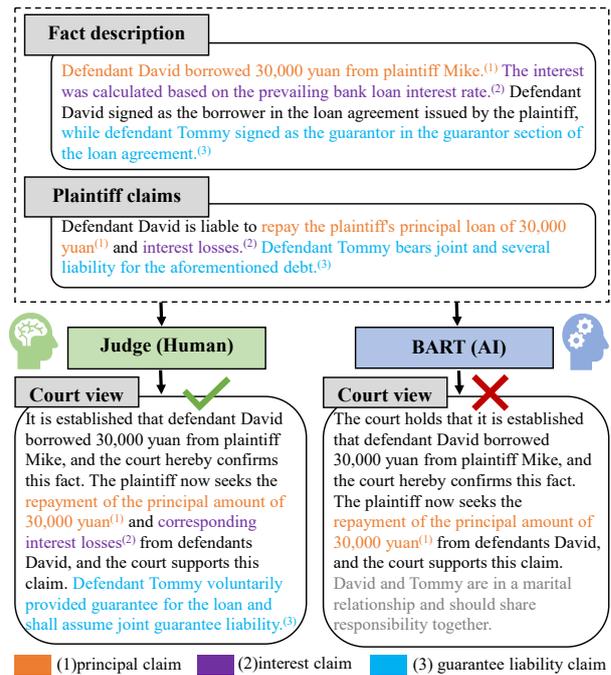


Figure 1: A real case in CVG. BART does not respond to the plaintiff's interest claim and mistakenly responds to the guarantee liability claim.

generated court view lacking the response to the interest claim. Similarly, BART also provides an incorrect response to the guarantee liability claim. In conclusion, general NLG methods may produce incomplete or erroneous court views, due to a lack of emphasis on domain-specific knowledge.

In this paper, we focus on the knowledge problem in the CVG task. To address this problem, we are still facing the following challenges: (1) Fine-

<sup>†</sup> Corresponding author.

tuning PLMs on domain-specific data consumes significant time and computational resources. How to inject the domain knowledge into the model efficiently? (2) There remains a gap in the model's acquisition and application of domain knowledge. How to guide the model in utilizing knowledge to fulfill the task's requirements?

To tackle the aforementioned challenges, based on the pretrained language model (PLM), we propose a Knowledge Injection and Guidance (KIG) method. To inject the domain knowledge efficiently, we propose a knowledge-injected prompt encoder, which focuses on training parameters within the prompt encoder itself. We inject two types of claim-related knowledge: keyword knowledge and label definition knowledge. Specifically, in the training stage, the knowledge-injected prompt encoder leverages keyword knowledge for its initialization. Additionally, it employs label definition knowledge to identify and emphasize claim-related information within the context. To guide the model in utilizing knowledge, we propose a generation navigator in the inference stage, which guides the PLM in generating text that incorporates useful domain knowledge. The navigator dynamically adjusts the decoding distribution, providing valuable guidance to the PLM in crafting appropriate responses to plaintiffs' claims. Importantly, it is worth noting that the navigator does not alter the architecture of the PLM and only is used in the inference stage, making it readily transferable to other models.

In order to evaluate the quality of generated court views more comprehensively, in addition to using similarity metrics, we design novel metrics to evaluate the model's ability to respond to claims. We conduct extensive experiments on a real-world dataset and demonstrate the effectiveness of our method through comparisons with multiple baselines (including ChatGPT). Our method has shown an improvement of 11.87% in the claim response metric compared to the best-performing baseline. In summary, our work's contributions can be summarized as follows:

- We investigate the task of court view generation (CVG) by taking the domain knowledge into consideration.
- We propose a Knowledge Injection and Guidance (KIG) method, incorporating a prompt encoder for knowledge injection and a navigator for knowledge utilization guidance. The navigator holds transferability on other PLMs.
- We design claim response metrics for the CVG task. Experimental results on a real-world dataset demonstrate the effectiveness of our method by both automatic evaluation and human evaluation. We make the code and

dataset publicly available <sup>1</sup> for reproducibility.

## 2. Related Work

### 2.1. Prompt Learning

Prompt learning is an approach that augments the input with prompt information to guide the model in adapting to downstream tasks. Researchers demonstrated its effectiveness using manually designed prompts that were intuitive and straightforward (Reynolds and McDonnell, 2021; Yu et al., 2022). However, manual prompt construction can be time-consuming and labor-intensive, leading to the development of automatic methods (Jiang et al., 2020; Shin et al., 2020). These methods all use discrete prompts. In continuous prompt methods, they inserted trainable parameters into the input and constructed prompts in the vector space (Li and Liang, 2021; Liu et al., 2021a,b; Gu et al., 2021). Additionally, studies have shown that using task-related token (Li and Liang, 2021) or prior knowledge (Chen et al., 2022) to initialize the prompt can enhance the performance of some tasks. In contrast to prior work's emphasis on performance in low-resource situations, our focus is on efficiently injecting domain knowledge to tackle domain-specific problems by prompt tuning.

### 2.2. Controlled Text Generation

To make the generated text of PLMs satisfy certain constraints or serve specific purposes, controlled text generation (CTG) techniques have rapidly developed. Some researchers believe that achieving CTG requires retraining or modifying the model (Keskar et al., 2019). Other researchers have achieved control effects by fine-tuning PLMs on specific corpora (Zhang et al., 2020; Ziegler et al., 2020; Zhang and Song, 2022). Additionally, some researchers proposed that CTG can be achieved through straightforward post-processing techniques (Yang and Klein, 2021), where the model's output is modified during the inference stage. In our work, we employ a generating navigator to ensure that generated text adheres to the attribute of "accurate knowledge utilization", which is similar to CTG. Unlike existing methods, we recognize the impact of generation length and design a dynamic guiding mechanism.

### 2.3. Court View Generation

The task of court view generation (CVG) has spurred the development of different methods across various scenarios. Yue et al. (2021b) divided the factual description into the adjudging cir-

<sup>1</sup><https://github.com/LIANG-star177/KIG>

Claim Labels	Keywords	Descriptions
Principal Claim	{“principal”, “debt”, “borrower”}	The requests for repayment of the initial borrowed or owed amount, excluding interest and additional charges.
Interest Claim	{“interest”, “interest rate”, “bank”}	A borrower requests to pay the interest on a owed amount, calculated based on the agreed-upon interest rate in the loan contract.
Spousal Joint Debt Claim	{“spouse”, “joint debt”, “property division”, “marriage”}	One spouse seeks to divide shared debts within a marriage, often occurring during divorce or separation when assets and debts are being split.
Guarantee Liability Claim	{“guarantor”, “guarantee”, “guaranty contract”}	A guarantor asks to fulfill their duties in a guarantee contract, typically because the borrower failed to meet their contract terms, leading to the guarantor paying the debt or fulfilling guaranteed responsibilities.

Table 1: Keywords and Label definitions of each claim label.

cumstance and the sentencing circumstance. They subsequently predicted the crime, generated rationales, and ultimately combined them to form the court view. Wu et al. (2020) introduced an attention mechanism specifically designed for plaintiff claims. They also proposed a pair of counterfactual decoders, which categorize the claims into supported and unsupported, intending to highlight the significance of claims. While these works adapt the model structure to fit CVG, our approach achieves this by incorporating domain knowledge, making it better suited to the specific requirements of CVG.

### 3. Problem Formulation

Here, we formalize our problem as follows:

**Fact Description** refers to a descriptive text that represents events that have been conclusively identified by the court. Here, we denote fact description  $f = \{w_1^f, \dots, w_{l_f}^f\}$ , where  $l_f$  denotes its length.

**Plaintiff Claims** refer to the specific allegations and requests made by the plaintiff against the defendant. Here, we denote plaintiff claims  $c = \{w_1^c, \dots, w_{l_c}^c\}$ , where  $l_c$  denotes its length.

**Claim Labels** are categories into which plaintiff claims can be summarized. We use  $m$  claim labels to encompass all the claims. We extract these labels from the plaintiff claims through word matching according to keywords<sup>2</sup> in Tab. 1.

**Court View** consists of the corresponding response to the plaintiff claims and a comprehensive analysis of fact description. We denote court view  $v = \{w_1^v, \dots, w_{l_v}^v\}$ , where  $l_v$  denotes its length.

Then, our task could be denoted as

**Problem 1 (Court View Generation)** *Given the fact description  $f$  and the plaintiff claims  $c$ , the task is to generate the court view  $v$ .*

## 4. Method

In this section, we first introduce the prompt tuning of pretrained language model (PLM), which

<sup>2</sup>We ask legal experts to supply these keywords.

serves as the foundational model. Then, we provide the detailed implementation of our Knowledge Injection and Guidance (KIG) method. The overall framework is shown in Fig. 2.

### 4.1. Preliminary

**Prompt Tuning.** Li and Liang (2021) train a prompt encoder that can provoke the PLM’s generation ability instead of fine-tuning. Specifically, in the CVG task, a prefix  $p$  of length  $l_p$  is concatenated in front of the fact description, plaintiff claims, and court view denoted as  $x = [p, f, c, v]$ . The embeddings of the prefix are randomly initialized and denoted as  $\mathcal{D}_p = [e(x_1), \dots, e(x_{l_p})]$ . Then, a trainable feedforward neural network ( $MLP_\theta$ ) serves as a prompt encoder, transforming the prefix into the sequence of hidden states. The hidden states, excluding those from the prefix, are obtained through encoding by a pretrained autoregressive language model ( $PLM_\phi$ ). The activation at the  $i$ -th time step  $h_i$  can be calculated as follows:

$$h_i = \begin{cases} MLP_\theta(\mathcal{D}_p)[i, :], & \text{if } i < l_p, \\ PLM_\phi(x_i, h_{<i}), & \text{otherwise,} \end{cases} \quad (1)$$

While encoding with the PLM,  $h_i = [h_i^1, \dots, h_i^n]$  is computed by concatenating all activation layers.  $n$  is the number of Transformer layers. Then, the probability distribution of each token in the target can be calculated as follows:

$$P(x_{i+1}|h_{\leq i}) = \text{softmax}(W_\phi h_i^n) \quad (2)$$

where  $W$  is a pretrained matrix that maps  $h_i^n$  to logits over the vocabulary.

Despite the frozen parameters in the PLM, the activations introduced by the prefix naturally influence the subsequent activations of the PLM, enabling prompt tuning to achieve notable performance in low-resource scenarios. To leverage the efficiency offered by prompt tuning and address the lack of domain knowledge, we introduce a knowledge-injected prompt encoder.

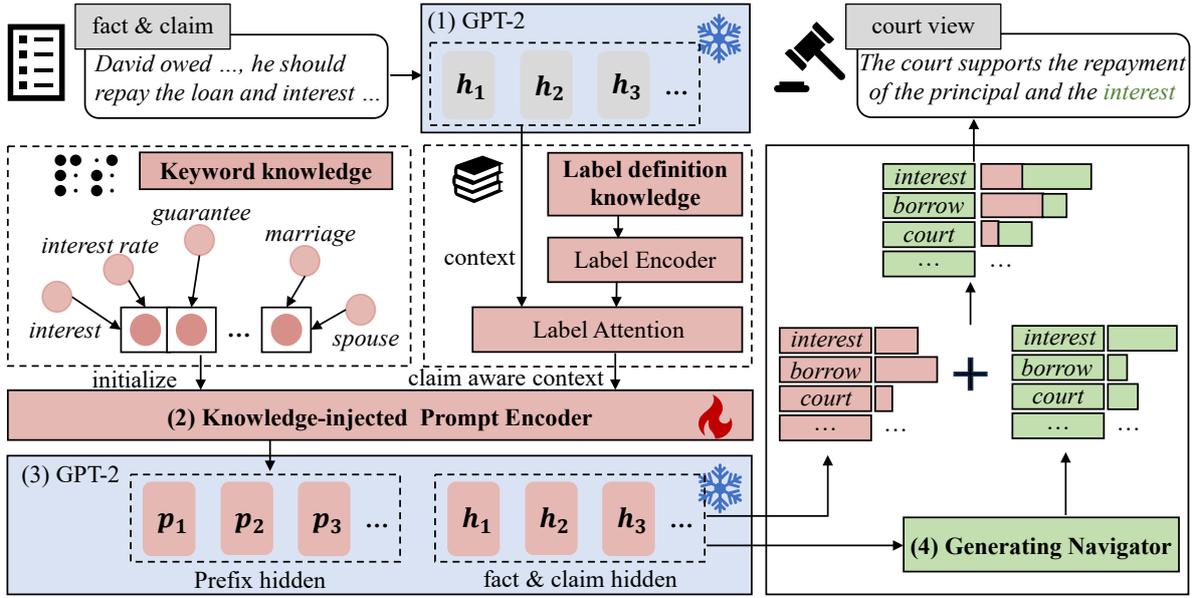


Figure 2: The architecture of KIG. (1) is the frozen pretrained GPT-2, utilized for obtaining input context. (2) is the knowledge-injected prompt encoder, which incorporates claim knowledge to obtain claim-aware context. (3) is the same frozen GPT-2, which is used for calculating the activation of output from the knowledge-injected context. (4) is the generating navigator, guiding the model's inference process.

## 4.2. Knowledge-injected Prompt Encoder

To efficiently address the CVG task, we incorporate two types of claim-related knowledge into the prompt encoder: keyword knowledge for initialization and label definition knowledge for label attention. The keywords and label definitions for each claim label are provided in Tab. 1.

### 4.2.1. Keyword-based Initialization

To enable the model to capture fine-grained claim information, we design a method to inject knowledge into the initialization of prompt embedding. Specifically, we first extract the representations of keywords in PLM's embedding layer and then aggregate these representations of each claim label as follows:

$$\hat{e}_i = \sum_i^K \phi_i^{cl} \cdot E(S_i), \quad (3)$$

where  $K$  is the key words' number of the  $i$ -th claim label and  $E$  is the word-embedding layer of the PLM.  $S_i$  is the set of keywords for the  $i$ -th claim label, while  $\phi_i^{cl}$  is the frequency distribution of these keywords. These claim label embeddings are then concatenated to initialize the embedding of the prompt encoder  $\mathcal{D}_p = [\hat{e}_1, \dots, \hat{e}_m]$ . So the prefix length  $l_p$  is equal to the number of claim labels  $m$ .

### 4.2.2. Label Attention

We notice that injecting semantic knowledge of claim labels can enhance the model's ability to learn claim-related information. Since every claim

label has a detailed definition in the Code of Law, we encode them using Transformer networks and a mean pooling operation to obtain the initial semantic claim label representation  $C \in \mathbb{R}^{m \times d}$ .

To allow the prompt encoder to sense different contexts, we concatenate the fact description  $f$  and the plaintiff claims  $c$  and inputs them into the PLM to obtain the context hidden  $h_{fc} \in \mathbb{R}^{(l_f+l_c) \times d}$ . We use the context hidden to query the relevant semantics of the claim label representation. Specifically, we first calculate the similarity score between the context hidden  $h_{fc}$  and the  $i$ -th claim label representation  $C_i$  as follows:

$$\alpha_i = h_{fc}^T W_c C_i, \quad (4)$$

where  $W_c$  is a trainable matrix. We calculate the weighted semantic representations of claim labels based on the corresponding similarity scores:

$$C' = \sum \frac{\exp(\alpha_i)}{\sum_{j=1} \exp(\alpha_j)} C_i \quad (5)$$

We further add the input's context hidden with the most relevant semantic representation of the claim labels, obtaining the claim-aware context hidden:

$$h'_{fc} = h_{fc} + C' \quad (6)$$

Since the MLP in prompt tuning (Li and Liang, 2021) can only encode prefixes, in order to combine both claim-aware context and prefix information simultaneously, we replace the MLP with an autoregressive model  $R_\theta$  as the prompt encoder.

It can be expressed as follows:

$$h'_i = \begin{cases} R_{\theta}(\hat{\mathcal{D}}_p)(x_i, h'_{fc}, h_{<i}), & \text{if } i < l_p, \\ PLM_{\phi}(x_i, h_{<i}), & \text{otherwise.} \end{cases} \quad (7)$$

Therefore, we predict the probability distribution of each token similar to Eq. 2.

### 4.3. Generating Navigator

In the inference stage, we employ a generating navigator to guide the utilization of domain knowledge. Specifically, it adjusts the decoding probability distribution of the PLM to generate court views that incorporate appropriate knowledge. We will now explain the principles and the guiding process.

#### 4.3.1. Principles

Inspired by Yang and Klein (2021), we explain the principles as follows:

If the generative model needs to generate a court view  $v$  of length  $l_v$ , with the attribute  $a$ , it can be modeled as follows:

$$P(v|a) = \prod_{i=1}^{l_v} P(x_i|x_{1:i-1}, a), \quad (8)$$

where  $x_i$  is the  $i$ -th token, we can utilize Bayesian decomposition to obtain:

$$P(x_i|x_{1:i-1}, a) \propto P(x_i|x_{1:i-1})P(a|x_{1:i}), \quad (9)$$

where  $P(x_i|x_{1:i-1})$  is the distribution of the next token.  $P(a|x_{1:i})$  can be modeled using a classifier that predicts the attribute  $a^3$ .

Therefore, we can employ a claim label classifier as the generating navigator to assign the recommendation score for the next token and its training details can be found in Sec. 4.4.

#### 4.3.2. Recommendation Score Calculation

Now we present the calculation process of the recommendation score. Given a generated sequence  $x_{1:i-1}$ , to obtain the navigator's recommendation score distribution  $\phi_i^s$  for  $i$ -th token, we evaluate the normalized Jaccard similarity between the predicted label  $a$  and the true label  $\hat{a}$  as follows:

$$\phi_i^s = \text{softmax}\left(\frac{|a_i \cap \hat{a}|}{|a_i \cup \hat{a}|}\right). \quad (10)$$

Moreover, we consider the impact of the generation length. When the generation length is short, the navigator receives little information, resulting in poor performance and unreliable guiding capability. We design a dynamic guidance method based

on the generation length  $l$ , and the new scoring calculation method is as follows:

$$\tilde{\phi}_i^s = \frac{\phi_i^s}{1 + \exp((k-l)/\mu)}, \quad (11)$$

where  $k$  denotes the starting position of the guidance,  $\mu$  controls the variation speed of the guiding strength. As the generation length  $l$  increases, the guiding strength becomes stronger.

#### 4.3.3. Inference with Guidance

In the inference stage, we concatenate the fact description and the plaintiff claim together, obtaining a new input for the model to calculate the next token's distribution  $\phi_i^g$ . Simultaneously, we employ the generating navigator to calculate the recommendation score distribution  $\tilde{\phi}_i^s$ . Then, we aggregate the two distributions to yield the desired distribution  $\phi_i^g$  for  $i$ -th token:

$$\tilde{\phi}_i^g = \phi_i^g + \lambda \tilde{\phi}_i^s, \quad (12)$$

where  $\lambda$  is the guiding strength coefficient. Subsequently, we obtain the court view by selecting the token with the highest probability distribution.

### 4.4. Training

**Training for Knowledge-injected Prompt Encoder.** We keep the PLM's parameters frozen while training the prompt encoder to make the generated texts align with the real court views. Our training objective aims to minimize the following loss function:

$$\mathcal{L}(\theta) = - \sum_{i=l_f+l_c+1}^N \log p(\phi, \theta)(x_i|h_{<i}), \quad (13)$$

where  $\phi$  is the frozen parameters of the PLM and  $\theta$  is the trainable parameters of the prompt encoder.

**Training for generating navigator.** Suppose we have a dataset  $\{(x_{1:l_v}, \hat{y})\}$ , where  $x_{1:l_v}$  is a court view, and  $\hat{y}$  is the corresponding claim label. Since the navigator needs to predict at each time step during the decoding process, we separate each sample  $x_{1:l_v}$  into individual sub-samples  $x_{1:i}$ .

We train a multi-label classifier to predict the claim labels and compare the predicted label  $y$  with  $\hat{y}$ . We use cross-entropy loss (Nam et al., 2014) for optimizing the loss in multi-label classification:

$$\mathcal{L}_N = - \sum_{i=1}^m (y_i \log(\hat{y}_i)) + (1 - y_i) \log(1 - \hat{y}_i), \quad (14)$$

where predicted label for the  $i$ -th claim is  $y_i \in [0, 1]$ , the true label is  $\hat{y}_i \in [0, 1]$ , and  $m$  is the number of claim labels. We utilize GPT-2 to train the multi-label classifier, enabling us to compute and optimize gradients for all sub-samples in a single forward-backward process.

<sup>3</sup>In our task,  $a$  is including the correct claim label.

Type	Result
# Sample	41693
Avg. # Tokens in Fact Description	177.5
Avg. # Tokens in Plaintiff Claims	76.3
Avg. # Tokens in Court View	205.7
Avg. # numbers of claim labels	2.13

Table 2: Statistics of the dataset.

## 5. Experiment

### 5.1. Dataset Description

We conduct experiments on a public civil cases dataset (Wu et al., 2020). To ensure data quality, we set the minimum length for fact description, plaintiff claim, and court view to 20, while the maximum lengths are set to 400, 200, and 400, respectively. We randomly split the dataset into training, validation, and testing sets in an 8:1:1 ratio. The original dataset does not contain claim labels and we extract them using word matching, as shown in the Sec. 3. The details of the dataset are presented in Tab. 2.

### 5.2. Evaluation Metrics

#### 5.2.1. Automatic Evaluation

**Similarity metrics.** We use two commonly used metrics in NLG tasks. (1) **BLEU** (Papineni et al., 2002), measures the extent to which generated content matches the n-grams of the reference text. We used BLEU-1, BLEU-2 and BLEU-n, which is the average value of BLEU-1  $\sim$  4. (2) **ROUGE**<sup>4</sup>, is another evaluation metric for text generation. In particular, we utilized ROUGE-L (Lin, 2004) to measure the longest common subsequence (LCS).

**Claim response metrics.** In the Controlled Text Generation (CTG) task, Yang and Klein (2021) trained a classifier to predict the formality of translated text. Similarly, we train a claim label classifier on the training data to evaluate whether the generated court views accurately respond to the plaintiff claims. Specifically, we input the court views generated by the model into the classifier and obtain predicted claim labels. After statistics, the Micro-F1 and Macro-F1 of the classifier are 95.32% and 94.38%, respectively, which illustrates the classifier can effectively compute claim response metrics. We use the F1 score (Mi-F, Ma-F) and Jaccard similarity coefficient (Mi-J and Ma-J) to measure the alignment between predicted labels and true labels, where Mi refers to the micro average and Ma refers to the macro average.

<sup>4</sup><https://pypi.org/project/rouge/>

### 5.2.2. Human Evaluation

To further evaluate the effectiveness of KIG, we conduct a human evaluation. We randomly select 200 civil cases from the test set and shuffle them to ensure fairness. We invite ten annotators with legal backgrounds to evaluate these results refer to the ground truth from two perspectives: (1) **Fluency**. The annotators rate the fluency of court views on a scale of 1-5. (2) **Fitness**. The annotators assess the alignment between the court view and the plaintiff claims assigning a score between 1-5.

### 5.3. Baseline Methods

First, we implement several NLG methods that are trained from scratch. (1) **Transformer** (Vaswani et al., 2017), is a common method that leverages self-attention mechanisms. (2) **PGN** (See et al., 2017), is a sequence-to-sequence model that designs a pointer mechanism to handle out-of-vocabulary words. (3) **AC-NLG** (Wu et al., 2020) is a civil CVG method, that mitigates the bias of data.

Recently, there has been a surge in the popularity of large language models (LLMs). We incorporate a prompt (“Please generate a court view based on the following facts and the plaintiff claims”) before the input and utilize their interfaces to generate court views. (4) **Text-Davinci-003**, can generate high-quality text outputs with its large-scale pre-training on diverse text data. (5) **GPT-3.5-Turbo**<sup>5</sup>, is an upgraded version of the GPT-3 model, and exhibits impressive NLG capabilities.

In addition, we implement several PLMs-based methods. (6) **GPT-2**, is fine-tuning the pretrained GPT-2<sup>6</sup> model on civil court cases. (7) **BART** (Lewis et al., 2019) is a bidirectional autoregressive Transformer model that performs well in various NLP tasks. (8) **T5** (Raffel et al., 2020) is a text generation model based on the Transformer architecture and follows a text-to-text transfer learning paradigm. (9) **Prefix-Tuning** (Li and Liang, 2021) is a prompt learning method that employs a prompt encoder to encode prefixes. We use the same pretrained GPT-2 model as the base model to implement this method.

Lastly, we conduct ablation experiments on our approach. (10) **KIG w/o V** removes the knowledge prompt initialization from the prompt encoder and instead initializes it with random embeddings. (11) **KIG w/o L** removes the label attention from the prompt encoder and uses the context hidden of the input. (12) **KIG w/o N** removes the generating navigator in the inference stage, generating court views solely based on the decoder.

<sup>5</sup>GPT-3.5-Turbo is the model that powers ChatGPT

<sup>6</sup><https://huggingface.co/uer/gpt2-chinese-cluecorpussmall>

Method	Similarity Metrics						Claim Response Metrics			
	B-1	B-2	B-N	R-1	R-2	R-L	Mi-F	Ma-F	Mi-J	Ma-J
Transformer	61.01	52.01	48.97	64.47	46.45	56.95	72.02	65.24	56.28	50.86
PGN	61.24	51.25	47.68	68.16	46.16	58.09	67.01	64.96	52.01	48.59
AC-NLG	61.72	52.61	49.71	68.57	49.46	59.97	69.03	66.03	54.70	51.59
Text-Davinci-003	50.02	35.29	29.11	50.87	20.32	30.39	73.26	62.55	57.80	49.34
GPT-3.5-Turbo	53.47	39.38	32.86	53.25	23.38	32.46	79.23	64.76	65.60	53.23
BART	64.48	55.72	51.75	72.12	51.11	61.89	76.23	71.47	61.59	58.54
T5	63.33	53.63	50.33	69.21	46.39	61.70	74.01	72.36	58.74	59.25
GPT-2	<u>68.65</u>	<u>61.92</u>	<u>59.05</u>	<u>75.87</u>	<u>59.06</u>	<u>68.48</u>	<u>81.09</u>	<u>75.91</u>	<u>68.19</u>	63.20
Prefix-Tuning	66.28	58.36	57.12	73.95	56.52	65.69	79.28	75.24	67.59	<u>63.24</u>
KIG	<b>71.04</b>	<b>64.80</b>	<b>62.40</b>	<b>77.28</b>	<b>62.25</b>	<b>71.11</b>	<b>90.21</b>	<b>87.78</b>	<b>82.16</b>	<b>78.40</b>
KIG w/o V	70.48	63.58	61.06	76.47	60.85	69.99	88.29	83.65	79.04	72.91
KIG w/o LA	69.32	63.16	60.86	75.89	60.68	69.85	88.63	86.48	79.58	76.39
KIG w/o N	70.45	64.34	61.96	77.05	62.09	70.96	88.08	85.04	78.70	74.22

Table 3: Results of court view generation, the best is **bolded** and the second best is underlined.

Method	Similarity Metrics		Claim Response Metrics	
	B-N	R-L	Mi-F	Ma-F
Transformer w/ N	48.58(-0.39)	56.63(-0.32)	73.29(+1.27)	66.54(+1.30)
PGN w/ N	46.95(-0.73)	57.62(-0.47)	68.49(+1.38)	65.53(+0.57)
AC-NLG w/ N	49.51(-0.20)	59.57(-0.40)	70.24(+1.21)	67.42(+1.39)
BART w/ N	51.98(+0.23)	62.99(+1.10)	78.42(+2.19)	72.54(+1.07)
T5 w/ N	51.10(+0.77)	62.37(+0.67)	75.80(+1.79)	73.95(+1.59)
GPT-2 w/ N	60.31(+1.26)	69.26(+0.78)	83.55(+2.46)	79.02(+3.11)
Prefix-Tuning w/ N	58.66(+1.54)	67.33(+1.64)	81.44(+1.76)	77.05(+1.81)

Table 4: Results of the baselines with the generating navigator.

Method	Flu.	Fit.
AC-NLG	3.18	2.52
GPT-3.5-Turbo	3.86	3.82
GPT-2	4.49	3.87
KIG	<b>4.52</b>	<b>4.02</b>

Table 5: Results of human evaluation.

Model	Tuned Params	Storage Space	Training Time
Finetuning	118M	464M	1.4 hours
KIG	18M	72M	0.5 hours

Table 6: Efficiency Comparison of different methods on GPT-2. The training time is tested on 2 3090 GPUs.

## 5.4. Experiment Details

In prompt initialization, we set the prefix length to the number of claim labels. To maintain alignment, we set the same prefix length in the prefix-tuning method. When using the generating navigator to guide the generation process, we empirically set the range for changing the probability distribution to 10 (Yang and Klein, 2021). This means that the top 10 tokens with the highest probabilities are scored by the generating navigator and re-ordered.

We explore different hyperparameters of our model, and our work reports the results obtained with the best-performing parameters, where  $\lambda$  is 6,  $k$  is 50, and  $\mu$  is 10. To assess the efficiency of our KIG method, we compared the training costs between fine-tuning and KIG in Tab. 6.

## 5.5. Experimental Results

**Results of court view generation:** Tab. 3 shows results of CVG in automatic metrics. We have the following observations: (1) AC-NLG performs the best among the methods trained from scratch, but the PLMs-based methods achieve better results, with GPT-2 demonstrating the best performance among all baselines. (2) Our method outperforms the baselines across all metrics. KIG outperforms GPT-2 by 3.35% in the B-N and by 11.87% in the Ma-F. (3) The interface-based methods perform poorly in similarity metrics but excel in claim response metrics, which demonstrates that while LLMs may lack legal knowledge, they possess exceptional comprehension abilities. (4) Compared to GPT-2, prefix-tuning performs worse, which proves that simply replacing fine-tuning with prompt learning leads to a performance decline. (5) In the ablation experiments, KIG w/o V, KIG w/o LA, and KIG w/o N perform worse than KIG in all metrics,

Fact Description	Defendant David borrowed 30,000 yuan from plaintiff Mike. <sup>(1)</sup> The interest was calculated based on the prevailing bank loan interest rate. <sup>(2)</sup> Defendant David signed as the borrower in the loan agreement issued by the plaintiff, while defendant Tommy signed as the guarantor in the guarantor section of the loan agreement. <sup>(3)</sup> Subsequently, despite the plaintiff's repeated demands for payment, both defendants failed to make any payments, leading to the initiation of this lawsuit.
Plaintiff's Claims	Defendant David is liable to repay the plaintiff's principal loan of 30,000 yuan <sup>(1)</sup> and interest losses. <sup>(2)</sup> Defendant Tommy bears joint and several liability for the aforementioned debt. <sup>(3)</sup>
Court's View (BART)	The court holds that defendant David borrowed 30,000 yuan from plaintiff Mike, and the court hereby confirms this fact. The plaintiff now seeks the repayment of the principal amount of 30,000 yuan <sup>(1)</sup> and corresponding interest losses <sup>(2)</sup> from defendants David, and the court supports this claim. David and Tommy are in a marital relationship and should share responsibility together. <sup>(4)</sup>
Court's View (KIG)	The court has determined that defendant David indeed obtained a loan of 30,000 yuan from plaintiff Mike, and the court affirms this established fact. The plaintiff now requests the return of the principal sum of 30,000 yuan <sup>(1)</sup> along with the associated interest losses <sup>(2)</sup> from defendant David, and the court upholds this demand. Defendant Tommy, of his own accord, offered a guarantee for the loan and shall share joint liability for this guarantee. <sup>(3)</sup>
Court's View (REAL)	The court holds that it is established that defendant David borrowed 30,000 yuan from plaintiff Mike, and the court hereby confirms this fact. The plaintiff now seeks the repayment of the principal amount of 30,000 yuan <sup>(1)</sup> and corresponding interest losses <sup>(2)</sup> from defendant David, and the court supports this claim. Defendant Tommy voluntarily provided a guarantee for the loan and shall assume joint guarantee liability. <sup>(3)</sup>

(1) principal claim
  (2) interest claim
  (3) guarantee liability claim
  (4) spousal joint debt claim

Figure 3: Case study.

demonstrating the effectiveness of our approach. Specifically, knowledge prompt Initialization significantly improves the response metric, while label attention noticeably enhances the similarity metric.

**Results of the baselines with the generating navigator:** From Tab. 4, we can conclude that: (1) The generating navigator improves the claim response metrics across all baselines, indicating its versatility and transferability. (2) In terms of the similarity metric, the generating navigator progressively improves the PLMs-based methods but slightly decreases the methods trained from scratch. We believe this is because the navigator is the PLM architecture and is better aligned with the PLM generator. In summary, the generating navigator can be easily transferred to other models to meet the requirements of specific tasks.

**Results of human evaluation:** From Tab. 5, it can be seen that: (1) Due to the lack of rich language knowledge, AC-NLG performs worse than methods that involve PLMs. (2) Our method KIG achieves the best performance in fluency and fitness, which is consistent with the results in the automatic evaluation. (3) The Kappa coefficient  $\kappa$  between any two annotators is greater than 0.8, indicating the consistency of the human evaluation.

From the above observations, our method can generate high-quality court views. Specifically, it can provide effective responses to the plaintiff claims while aligning with true court views.

**The impact of hyperparameters on performance:** From the upper of Fig. 4, it can be observed that as the coefficient  $\lambda$  increases, the claim response metrics gradually improve, but the similarity metrics start to decrease after  $\lambda = 6$ . This indicates that excessive adjustment by the generating navigator leads to a decrease in performance. Therefore, in our work, we set  $\lambda$  to 6. From the lower, we can find that when  $k$  exceeds 50, there

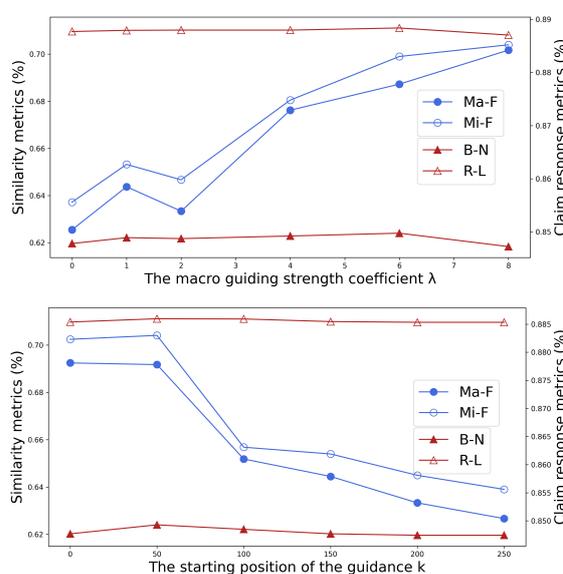


Figure 4: The upper shows the impact of guiding strength coefficient  $\lambda$  and the lower shows the impact of the guidance starting position  $k$ .

exists a noticeable decline in the guidance ability of the generating navigator. This suggests that the court view starts responding to the claims at around 50 tokens. In our work, we set  $k$  to 50.

## 5.6. Case Study

Fig. 3 displays court views in a real legal case, generated by the PGN, BART, and our method KIG, alongside the real court view. In this case, PGN failed to address the interest claim, while BART not only omitted a response to the interest claim but also incorrectly interpreted the guarantee liability claim as a spousal joint debt claim. In contrast, KIG accurately addressed all claims and maintained a similarity to the real court view.

## 6. Conclusion

In this paper, based on the PLM, we propose an efficient Knowledge Injection and Guidance (KIG) method for the CVG task. Our method integrates claim-related knowledge by prompt tuning and introduces a generating navigator to guide the generation process. Through comprehensive experiments on a real-world dataset, we demonstrate the effectiveness of our method in generating court views. Our approach is based on GPT-2 but does not modify the structure of GPT-2, so our method has the potential to be generalized to other LLMs, such as ChatGLM<sup>7</sup>, which also use prompt tuning.

## 7. Ethical Discussion

It is crucial to consider the ethical implications of CVG using artificial intelligence. First, Our method is constructed on a Chinese legal scene, and if it is to be applied to other countries' legal systems, adaptive adjustments need to be made. When dealing with a large amount of legal data, which involves personal privacy, it's crucial to adhere to privacy protection regulations (Xu et al., 2023). Another concern is that models can inadvertently perpetuate biases from training data. Therefore, it is essential to mitigate model biases to ensure fairness.

Furthermore, while KIG can automate CVG, it is necessary to preserve the role of legal professionals in critically evaluating and interpreting the generated outputs. This work should serve as a tool to assist legal professionals rather than replace their expertise and judgment. So far, users and stakeholders should have a clear understanding of the potential limitations of this algorithm and it cannot be directly applied to real-world scenarios.

## 8. Limitations

In this section, we discuss the limitations of our work as follows:

- The KIG method enhances the understanding of litigation claims by injecting prior knowledge. To enhance performance, consideration could be given to appropriate feature extraction techniques (Wang et al., 2017). Moreover, law is a constantly evolving domain. Ensuring the timeliness of knowledge updates may pose challenges.
- The KIG method introduces a generating navigator to explicitly guide the generator's focus on claims. However, this explicit guidance often requires extra training and data.

## 9. Acknowledgements

This work was supported in part by National Key Research and Development Program of China (2022YFC3340900), National Natural Science Foundation of China (No. 62376243, 62037001, U20A20387), the StarryNight Science Fund of Zhejiang University Shanghai Institute for Advanced Study (SN-ZJU-SIAS-0010) and Project by Shanghai AI Laboratory (P22KS00111).

## 10. Bibliographical References

- Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019. [Neural legal judgment prediction in english](#). In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4317–4323. Association for Computational Linguistics.
- Xiang Chen, Ningyu Zhang, Xin Xie, Shumin Deng, Yunzhi Yao, Chuanqi Tan, Fei Huang, Luo Si, and Huajun Chen. 2022. Knowprompt: Knowledge-aware prompt-tuning with synergistic optimization for relation extraction. In *Proceedings of the ACM Web Conference 2022*, pages 2778–2788.
- 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)*, pages 648–664, Dublin, Ireland. Association for Computational Linguistics.
- Xiaodong Gu, Kang Min Yoo, and Sang-Woo Lee. 2021. Response generation with context-aware prompt learning. *ArXiv*, abs/2111.02643.
- Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. [How Can We Know What Language Models Know?](#) *Transactions of the Association for Computational Linguistics*, 8:423–438.
- Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. [Ctrl: A conditional transformer language model for controllable generation](#).
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

<sup>7</sup><https://github.com/THUDM/ChatGLM-6B>

- Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021a. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*.
- Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. Gpt understands, too. *arXiv preprint arXiv:2103.10385*.
- 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 2023, Taipei, Taiwan, July 23-27, 2023*, pages 1023–1034. ACM.
- 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*, pages 2727–2736, Copenhagen, Denmark. Association for Computational Linguistics.
- Jinseok Nam, Jungi Kim, Eneldo Loza Mencía, Iryna Gurevych, and Johannes Fürnkranz. 2014. [Large-scale multi-label text classification - revisiting neural networks](#). In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part II*, volume 8725 of *Lecture Notes in Computer Science*, pages 437–452. Springer.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.
- Laria Reynolds and Kyle McDonell. 2021. [Prompt programming for large language models: Beyond the few-shot paradigm](#).
- Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.
- Yunqiu Shao, Jiabin Mao, Yiqun Liu, Weizhi Ma, Ken Satoh, Min Zhang, and Shaoping Ma. 2020. Bert-pi: Modeling paragraph-level interactions for legal case retrieval. In *International Joint Conference on Artificial Intelligence*.
- Taylor Shin, Yasaman Razeghi, Robert L. Logan IV au2, Eric Wallace, and Sameer Singh. 2020. [Autoprompt: Eliciting knowledge from language models with automatically generated prompts](#).
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- 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*, pages 325–334. ACM.
- 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*, pages 485–494. ACM.
- Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. 2017. Community preserving network embedding. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI'17*, page 203–209. AAAI Press.
- 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)*, pages 763–780.
- Yiquan Wu, Siying Zhou, Yifei Liu, Weiming Lu, Xiaozhong Liu, Yating Zhang, Changlong Sun, Fei Wu, and Kun Kuang. 2023. [Precedent-enhanced legal judgment prediction with LLM and domain-model collaboration](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12060–12075, Singapore. Association for Computational Linguistics.

- Chaojun Xiao, Xueyu Hu, Zhiyuan Liu, Cunchao Tu, and Maosong Sun. 2021. [Lawformer: A pre-trained language model for chinese legal long documents](#). *AI Open*, 2:79–84.
- Jimin Xu, Nuanxin Hong, Zhening Xu, Zhou Zhao, Chao Wu, Kun Kuang, Jiaping Wang, Mingjie Zhu, Jingren Zhou, Kui Ren, Xiaohu Yang, Cewu Lu, Jian Pei, and Harry Shum. 2023. [Data-driven learning for data rights, data pricing, and privacy computing](#). *Engineering*, 25:66–76.
- 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*, pages 3086–3095. Association for Computational Linguistics.
- Kevin Yang and Dan Klein. 2021. [FUDGE: Controlled text generation with future discriminators](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics.
- Fangyi Yu, Lee Quartey, and Frank Schilder. 2022. [Legal prompting: Teaching a language model to think like a lawyer](#).
- Linan Yue, Qi Liu, Binbin Jin, Han Wu, Kai Zhang, Yanqing An, Mingyue Cheng, Biao Yin, and Dayong Wu. 2021a. [Neurjudge: A circumstance-aware neural framework for legal judgment prediction](#). In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 973–982.
- Linan Yue, Qi Liu, Han Wu, Yanqing An, Li Wang, Senchao Yuan, and Dayong Wu. 2021b. [Circumstances enhanced criminal court view generation](#). In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1855–1859.
- Hanqing Zhang and Dawei Song. 2022. [Discup: Discriminator cooperative unlikelihood prompting for controllable text generation](#).
- Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. 2022. [A survey of controllable text generation using transformer-based pre-trained language models](#).
- Yizhe Zhang, Siqi Sun, Michel Galley, Yen-Chun Chen, Chris Brockett, Xiang Gao, Jianfeng Gao, Jingjing Liu, and Bill Dolan. 2020. [Dialogpt](#). [Large-scale generative pre-training for conversational response generation](#).
- 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*, pages 3540–3549, Brussels, Belgium. Association for Computational Linguistics.
- Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2019. [Jec-qa: A legal-domain question answering dataset](#).
- Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. [How does NLP benefit legal system: A summary of legal artificial intelligence](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5218–5230, Online. Association for Computational Linguistics.
- Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2020. [Fine-tuning language models from human preferences](#).
- Xu Zou, Da Yin, Qingyang Zhong, Hongxia Yang, Zhilin Yang, and Jie Tang. 2021. [Controllable generation from pre-trained language models via inverse prompting](#). In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery Data Mining*. ACM.

## 11. Language Resource References

- 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)*, pages 763–780.