Law Article-Enhanced Legal Case Matching: A Causal Learning Approach

Zhongxiang Sun
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
sunzhongxiang@ruc.edu.cn

Jun Xu*
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
junxu@ruc.edu.cn

Xiao Zhang
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
zhangx89@ruc.edu.cn

Zhenhua Dong
Noah's Ark Lab, Huawei
Shenzhen, China
dongzhenhua@huawei.com

Ji-Rong Wen
Gaoling School of Artificial Intelligence
Renmin University of China
Beijing, China
jrwen@ruc.edu.cn

ABSTRACT

Legal case matching, which automatically constructs a model to estimate the similarities between the source and target cases, has played an essential role in intelligent legal systems. Semantic text matching models have been applied to the task where the source and target legal cases are considered as long-form text documents. These general-purpose matching models make the predictions solely based on the texts in the legal cases, overlooking the essential role of the law articles in legal case matching. In the real world, the matching results (e.g., relevance labels) are dramatically affected by the law articles because the contents and the judgments of a legal case are radically formed on the basis of law. From the causal sense, a matching decision is affected by the mediation effect from the cited law articles by the legal cases, and the direct effect of the key circumstances (e.g., detailed fact descriptions) in the legal cases. In light of the observation, this paper proposes a model-agnostic causal learning framework called Law-Match, under which the legal case matching models are learned by respecting the corresponding law articles. Given a pair of legal cases and the related law articles, Law-Match considers the embeddings of the law articles as *instrumental variables* (IVs), and the embeddings of legal cases as *treatments*. Using IV regression, the treatments can be decomposed into law-related and law-unrelated parts, respectively reflecting the mediation and direct effects. These two parts are then combined with different weights to collectively support the final matching prediction. We show that the framework is model-agnostic, and a number of legal case matching models can be applied as the underlying models. Comprehensive experiments show that Law-Match can outperform state-of-the-art baselines on three public datasets.

CCS CONCEPTS

• **Applied computing** → **Law**; • **Information systems** → **Content analysis and feature selection**.

KEYWORDS

Legal Case Matching, Causal Inference, Law

ACM Reference Format:
Zhongxiang Sun, Jun Xu, Xiao Zhang, Zhenhua Dong, and Ji-Rong Wen. 2023. Law Article-Enhanced Legal Case Matching: a Causal Learning Approach. 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, 10 pages. https://doi.org/10.1145/3539618.3591709

1 INTRODUCTION

Legal case matching has played an important role in intelligent legal systems. For example, in legal case retrieval, the matching models help the system to determine the relevance between the query cases and the candidate cases. Traditionally, the task is formalized as matching two long-form text documents at the semantic level. General-purpose document matching models have been adapted to tackle the problem, including the heuristic methods [27, 48], network-based methods [7, 8], and text-based methods [29, 41].

Though effective, simply considering the legal cases as general long-form text documents [45] still has spaces for improvement. One striking difference between legal cases and general documents is that legal cases usually cite a number of law articles[1]. These law articles are selected from the law book (e.g., Chinese Criminal Law) by the judges and provide essential knowledge of the legal case's context and judgments. Existing studies have shown that law articles are beneficial to a number of legal-related tasks [34, 43, 49].

*Jun Xu is the corresponding author. Work partially done at Engineering Research Center of Next-Generation Intelligent Search and Recommendation, Ministry of Education.

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

ACM ISBN 978-1-4503-9408-6/23/07…$15.00
https://doi.org/10.1145/3539618.3591709

[1]Law articles are the foundation of statutes or written laws which are usually enacted by the administration of justice (e.g., Criminal Law in China).

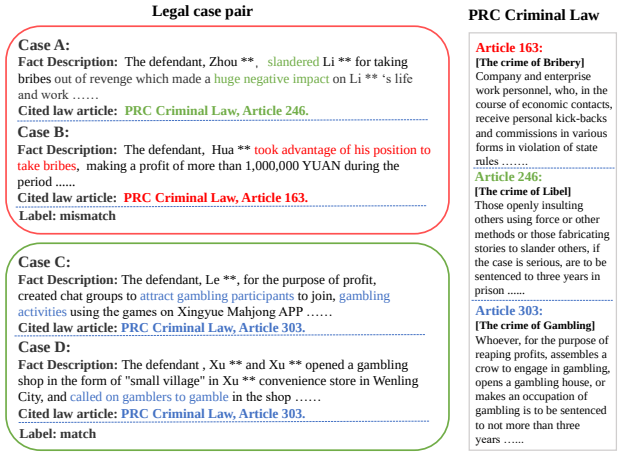


Figure 1: Left: two pairs of legal cases; Right: three cited law articles in the legal cases. (translated from Chinese)

Analysis shows that the law articles are also important in legal case matching. Figure 1 shows the snapshots of two real legal case pairs[2]. Contents of the cited law articles are listed in the right part of the figure. In the first legal case pair, Case A and Case B share a large number of words in their fact descriptions. However, the judges' decisions are: Case A is libel crime (PRC Criminal Law, Article 246) while Case B is bribery crime (Article 163). The associated law articles are helpful in identifying the key information (highlighted) in the two cases [14]. By comparing the key information in the two cases, experts annotate the matching label as "mismatch" though they have relatively high semantic text similarity (measured by Lawformer [41]). In the second example of Figure 1, Case C and Case D have relatively lower semantic text similarity than the previous pair. However, both of them are judged as the gambling crime (Article 303). The law article helps to identify similar key information (highlighted) in these two legal cases. So the expert-annotated matching label is "match".

Usually, the key constitutive elements and the key circumstances provide important signals for the matching of two legal cases [19]. The key constitutive elements are highly summative texts written in light of some law articles. The key circumstances, on the other hand, are detailed fact descriptions and are usually very different from case to case. They are not directly related to any law articles. Therefore, it is possible that law articles can help the matching model to identify and decompose these key information.

From the causal sense, the matching of two legal cases is affected by the *mediation effect* from the law articles and the *direct effect* from the key circumstances part of legal cases. More specifically, the key constitutive elements in the legal cases mediate the law articles' effect on the matching decision (i.e., the mediation effect). In contrast, the key circumstances directly affect the matching decision (i.e., the direct effect). As a result, the embedding of a legal case actually consists of two parts: the law-related part, which is the mediator of the mediation effect, and the law-unrelated part, which has direct effect. These two parts reflect different association mechanisms between the legal cases and the matching decisions. It is necessary to identify and treat them differently.

To address the issue, this paper proposes a causal representation learning framework tailored for legal case matching, called Law-Match. Specifically, Law-Match considers the legal cases as *treatment* and the corresponding law articles as *instrument variables* (IVs) [2, 9, 33, 38]. In the matching phase, after getting the embeddings of the legal cases (i.e., treatments) and the related law articles (i.e., IVs), Law-Match first uses the IVs to regress the treatments, resulting in the fitted vector (law-related part) and the residuals (law-unrelated part). These two parts have different effects on the final matching. Law-Match then combines them into a newly reconstructed treatment vector with the attention mechanism. Finally, the reconstructed treatment is fed to the underlying matching model for making the final matching prediction. In the training phase, an alternative optimization procedure is developed to learn the parameters in the IV regression and matching models.

We summarize the major contributions of the paper as follows:

(1) We analyze the essential role of law articles in legal case matching from a causal view: the matching decisions are affected by the mediation effect of the law articles and the direct effect of the key circumstances in the legal cases.

(2) We propose a novel model-agnostic causal learning framework which introduces the law articles into the process of legal case matching in a theoretically sound way. IV regression is adopted to decompose the mediation effect and direct effect from the legal case embeddings by considering law articles as IVs and legal cases as the treatments.

(3) We conducted extensive experiments on three public datasets. Experimental results demonstrated that Law-Match could significantly improve the underlying models' performance and outperform the baselines, verifying the importance of the law articles in legal case matching.

2 RELATED WORK

2.1 Legal case matching

Conventionally, legal case matching can be addressed with manual knowledge engineering (KE) [6]. The methods include the Boolean search technology and manual classification [11]. With the development of NLP, deep learning has been adapted to realize semantic level matching of legal cases. According to [8], these studies can be categorized as network-based and text-based methods. The network-based methods are tailored for common law and use the citations of different cases to construct a Precedent Citation Network (PCNet). For example, [17] use PCNet-based Jaccard similarity to infer the paired legal cases' similarity. Bhattacharya et al. [8] use Node2vec to map the nodes of the graph to a vector space and then compute the legal cases' cosine similarity. See also [7, 21].

The text-based methods compute the semantic similarity between legal cases. Shao et al. [29] utilize BERT to capture the semantic relationships at the paragraph level and then infer the relevance between two cases by aggregating the paragraph-level interactions. Xiao et al. [41] release the longformer-based [5] pre-trained language model to get a better representation of long legal documents. Yu et al. [46] propose a three-stage explainable legal case matching model. Law articles have shown their effects on a

[2]Crawled from http://faxin.cn and translated to English.

number of legal tasks. Zhong et al. [49] jointly model the law article prediction task and the Legal Judgment Prediction (LJP). Xu et al. [43] construct a relationship diagram between the law articles and introduce all relevant law articles into the LJP.

2.2 Causal learning

In causal learning, **instrument variable (IV)** [2, 9, 33, 38] has been widely used to identify the causal effect of the treatment on the output. Traditionally, two-stage least squares (2SLS) [1] regression is used to regress the IVs to treatments. Shaver [30] and Dippel et al. [12] use the linear 2SLS to identify both the causal treatment and mediation effects. Recently, models have been proposed to extend the linear 2SLS model to high dimensional and non-linear deep neural networks. Xu et al. [42] extend 2SLS to an alternating training regime and perform well in high-dimensional image data and off-policy reinforcement learning. See also [15, 22, 32, 40, 47].

Recently, **causal representation learning** has been proposed to discover the high-level causal variables from low-level observations [28]. Yang et al. [44] propose a VAE-based causal disentangled representation learning framework by leveraging labels of concepts as additional knowledge. Si et al. [31] reconstruct the causal representation of items in recommendation by using search data as the additional knowledge. The proposed Law-Match can also be viewed as learning causal legal case representations by using the law articles as additional knowledge.

Mediation analysis is designed to explore the underlying mechanism by which one variable influences another variable through a mediator variable [13]. In order to better leverage the different mechanisms, many studies are proposed to decompose the different effects [4, 37]. In this paper, we also use the IV regression to identify the indirect effects between law articles and matching results.

3 BACKGROUND AND PRELIMINARIES

3.1 Problem formulation

Suppose we are given a set of labelled data tuples $\mathcal{D} = \{(X, Y, z)\}$ where $X \in \mathcal{X}$ is a source legal case, $Y \in \mathcal{Y}$ is a target case, and $z \in \mathcal{Z}$ is the human-annotated matching label. $\mathcal{Z}$ could be defined as, for example, $\mathcal{Z} = \{0, 1, 2\}$ where 0 means mismatch, 1 means partially match, and 2 means match. Typically, a legal case can be considered as a sequence of words that describe the case's facts. Therefore, the legal case X (or Y) can be represented as a d-dimensional embeddings $\mathbf{e}(X) \in \mathbb{R}^d$ (or $\mathbf{e}(Y) \in \mathbb{R}^d$). Typically, the embeddings are the outputs at the [CLS] token of a BERT model pre-trained on a legal corpus.

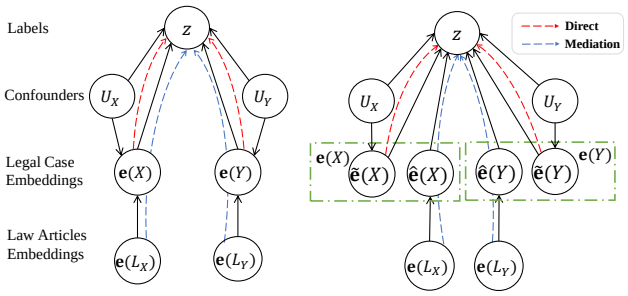
The task of legal case matching, therefore, becomes learning a matching model $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ based on the labelled tuples in $\mathcal{D}$.

3.2 Law articles in legal case matching

Real legal cases usually cite the applicable law articles that support the judicial decisions[3]. These law articles are selected from a law book (e.g., PRC Criminal Law). The IDs (i.e., IDs of the articles, clauses, and items) are listed at the end of legal cases. Therefore, we can collect the article contents from the law book according to the IDs. The law article contents can be concatenated as a new pseudo document, represented as another d-dimensional embedding $\mathbf{e}(L_X) \in \mathbb{R}^d$ where L_X denotes the law articles cited by X.

Intuitively, the law articles should provide complementary knowledge for understanding the legal cases and therefore enhancing the matching. On the one hand, legal cases are long-form documents containing multiple sentences, describing a number of facts. Some of them are the key facts, while others are not. The applicable law articles are selected by the judges. They should reflect the most key information (e.g., key facts) in the case, affecting the legal case matching. On the other hand, the law articles influence the description of the facts and the judgments (e.g., charges, terms of penalty). When preparing a legal case, the lawyers would consider the law articles seriously because the judge's decisions are based on the law articles.

One straightforward approach is concatenating the contents of the law articles to the original texts, i.e., appending L_X to X and appending L_Y to Y. Though improvements can be observed, we note that there exist fundamental differences between the law articles and texts in legal cases: the law articles are created by the governmental institutions and presented in the form of general rules with precise definitions. The legal cases are written by judges in the form of detailed descriptions of specific facts. They have different roles and affect the matching with different mechanisms.



(a) Causal graph of legal case matching. (b) Causal graph after IV regression.

Figure 2: Causal graph of legal case matching and the graph after IV regression. (a): Law article embeddings $\mathbf{e}(L_X)$ and $\mathbf{e}(L_Y)$ affect z through $\mathbf{e}(X)$ and $\mathbf{e}(Y)$, which also have other effects on z. (b): $\mathbf{e}(L_X)$ and $\mathbf{e}(L_Y)$ affect z through the fitted parts (mediators) $\widehat{\mathbf{e}}(X)$ and $\widehat{\mathbf{e}}(Y)$. The residuals $\bar{\mathbf{e}}(X)$ and $\bar{\mathbf{e}}(Y)$ have direct effects on z.

3.3 A causal view of legal case matching

Following the framework proposed in [23, 24], we can formalize legal case matching with a multivariate causal graph. According to Figure 2(a), the two input legal cases X and Y are *two treatment variables* in the causal graph, respectively represented as their embeddings $\mathbf{e}(X)$ and $\mathbf{e}(Y)$. The *outcome variable* z is the matching label. There exist associations between $\mathbf{e}(X)$ and z (path $\mathbf{e}(X) \to z$) and $\mathbf{e}(Y)$ and z (path $\mathbf{e}(Y) \to z$), because the prediction is based on the matching signals between X and Y.

[3] In some tasks such as legal retrieval, the source cases (queries) only contain the fact descriptions. For these cases, the law articles can be extracted with the causal discovery method [18]. Please refer to Section 5.1 for more details.

The observations in Ma et al. [19] show that the key constitutive elements in a legal case are generally highly related to the cited law articles. The key circumstances, however, are not. In a causal sense, the matching labels are determined along two paths, including the *mediation effect* of law articles and the *direct effect* of the key circumstances. More specifically, *the key constitutive elements mediate the effect of the law articles on the matching label*, while the key circumstances have direct effects on the matching label. Therefore, the associations $\mathbf{e}(X) \rightarrow z$ and $\mathbf{e}(Y) \rightarrow z$ are mixtures of two different types of causal paths, i.e., the law-related associations caused by the mediation effect and the law-unrelated associations caused by the direct effect.

Besides, for legal case X (or Y), there also exist missing variables U_X (or U_Y) that are associated with both X and z (paths $\mathbf{e}(X) \leftarrow U_X \rightarrow z$). The missing variables could be any confounding factors unrelated to law articles (e.g., the focus of disputes). However, they are important parts of the legal case and are considered when making the matching decisions. Therefore, the law-unrelated association can be viewed as a backdoor path in the causal graph. Obviously, the law-related and law-unrelated associations reflect different mechanisms between legal cases (i.e., treatments) and matching prediction (i.e., outcome). It is necessary to identify these two associations and treat them differently.

The independence between the law articles and the missing variables as well as the key circumstances, provide us a chance to conduct the identification. As shown in Figure 2(b), we leverage L_X and L_Y as the IVs [39, 42][4]. Thus, we can regress $\mathbf{e}(X)$ on $\mathbf{e}(L_X)$ and $\mathbf{e}(L_Y)$ to get $\widehat{\mathbf{e}}(X)$ which does not depend on the confounder U_X and residual $\widetilde{\mathbf{e}}(X) = \mathbf{e}(X) - \widehat{\mathbf{e}}(X)$. Therefore, path $\widehat{\mathbf{e}}(X) \rightarrow z$ can be viewed as purely law-related associations. Paths $\widetilde{\mathbf{e}}(X) \rightarrow z$ and $\widetilde{\mathbf{e}}(X) \leftarrow U_X \rightarrow z$ can be seen as totally law-unrelated. Similarly, we can regress the embedding of $\mathbf{e}(Y)$ on $\mathbf{e}(L_X)$ and $\mathbf{e}(L_Y)$, to get $\widehat{\mathbf{e}}(Y)$ which does not depend on the confounder U_Y, and the residual part $\widetilde{\mathbf{e}}(Y) = \mathbf{e}(Y) - \widehat{\mathbf{e}}(Y)$. $\widehat{\mathbf{e}}(Y) \rightarrow z$ can be viewed as law-related associations. $\widetilde{\mathbf{e}}(Y) \rightarrow z$ and $\widetilde{\mathbf{e}}(Y) \leftarrow U_Y \rightarrow z$ can be seen as law-unrelated associations. In this way, we can identify the law-related associations and law-unrelated associations under a causal framework.

4 OUR APPROACH: LAW-MATCH

This section presents an implementation of the causal learning framework for legal case matching, called Law-Match.

4.1 Model overview

Figure 3 illustrates the architecture of Law-Match. Given a pair of legal cases (X, Y), Law-Match first encodes them as two embeddings (two treatments). Also, the cited law articles L_X and L_Y are encoded as two embeddings (two IVs). Then, the treatment reconstruction module is employed to decompose each treatment into two vectors with the help of the corresponding IVs. After that, the decomposed vectors are combined as a new reconstructed vector. Finally, the reconstructed treatment vectors are fed to the downstream matching model for making the matching prediction.

[4]According to Wooldridge [39], $\widetilde{\mathbf{e}}(X)$ will contain an error term if we only use $\mathbf{e}(L_X)$ to regress $\mathbf{e}(X)$. We cannot control the association between $\widehat{\mathbf{e}}(X)$ and $\widetilde{\mathbf{e}}(X)$.

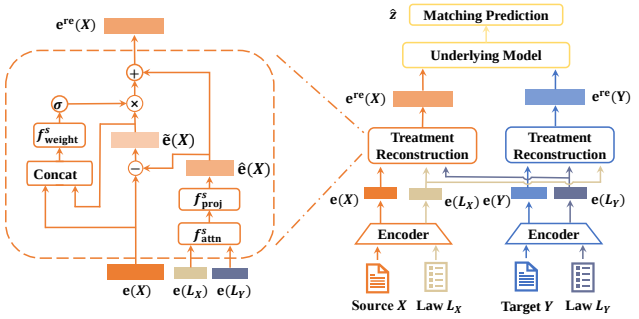


Figure 3: The architecture of Law-Match. Right: procedure of Law-Match applied to an underlying text matching model; Left: procedure of treatment reconstruction.

4.2 Treatment reconstruction

As shown in the left part of Figure 3, Law-Match employs two treatment reconstruction modules to process the source case X and target case Y, outputs the reconstructed embeddings $\mathbf{e}^{\text{re}}(X)$ and $\mathbf{e}^{\text{re}}(Y)$, respectively. These two modules share the same network architecture while with different parameters.

As have shown in Section 3.3, the new treatment $\mathbf{e}^{\text{re}}(X)$ can be created by first regressing $\mathbf{e}(X)$ on the IVs $\mathbf{e}(L_X)$ and $\mathbf{e}(L_Y)$, achieving the fitted part and residual part. We call this stage treatment decomposition. Then, these two parts are re-combined together with attended weights, called treatment reconstruction.

4.2.1 Treatment decomposition. IV regression is used to decompose $\mathbf{e}(X)$ into the fitted part $\widehat{\mathbf{e}}(X)$ and residual part $\widetilde{\mathbf{e}}(X)$, with the help of IVs $\mathbf{e}(L_X)$ and $\mathbf{e}(L_Y)$. Specifically, $\widehat{\mathbf{e}}(X)$ can be written as

$$\widehat{\mathbf{e}}(X) = f^s_{\text{proj}}\left(c_s(L_X, L_Y)\right), \tag{1}$$

where $f^s_{\text{proj}} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is a projection network that maps a law article embedding to the space of legal case embeddings, and the input vector $c_s(L_X, L_Y)$ is a linear combination of the two law article embeddings $\mathbf{e}(L_X)$ and $\mathbf{e}(L_Y)$:

$$c_s(L_X, L_Y) = w_s \cdot \mathbf{e}(L_X) + (1 - w_s) \cdot \mathbf{e}(L_Y),$$

where

$$w_s = \frac{\exp\{f^s_{\text{attn}}\left(\mathbf{e}(X), \mathbf{e}(L_X)\right)\}}{\exp\{f^s_{\text{attn}}\left(\mathbf{e}(X), \mathbf{e}(L_X)\right)\} + \exp\{f^s_{\text{attn}}\left(\mathbf{e}(X), \mathbf{e}(L_Y)\right)\}},$$

and $f^s_{\text{attn}}(\cdot, \cdot)$ denotes the additive attention [3]:

$$f^s_{\text{attn}}(\mathbf{a}, \mathbf{b}) = \mathbf{v}^{\text{T}} \tanh(\mathbf{W}[\mathbf{a}; \mathbf{b}]),$$

where $\mathbf{v}$ and $\mathbf{W}$ are learnable attention parameters and '$[\cdot;\cdot]$' denotes concatenation of two vectors. The fitted part $\widehat{\mathbf{e}}(X)$ reflects the law-related median association between the law article embeddings and the matching results.

Given $\widehat{\mathbf{e}}(X)$, it is easy to get the residual part:

$$\widetilde{\mathbf{e}}(X) = \mathbf{e}(X) - \widehat{\mathbf{e}}(X). \tag{2}$$

Obviously, $\widetilde{\mathbf{e}}(X)$ reflects the law-unrelated direct association between the legal case embeddings and the matching results.

4.2.2 Treatments reconstruction. The fitted parts and the residuals can be recombined, achieving a new treatment:

$$\mathbf{e}^{\text{re}}(X) = \widehat{\mathbf{e}}(X) + \alpha_s \cdot \widetilde{\mathbf{e}}(X), \tag{3}$$

where $\alpha_s \in [0, 1]$ re-weights the influence of the residual part:

$$\alpha_s = \sigma(f^s_{\text{weight}}([\mathbf{e}(X); \widetilde{\mathbf{e}}(X)])),$$

where f^s_{weight} denotes a two-layer MLP that takes the concatenation of the treatments and the residual part as input and outputs a real number, σ denotes the sigmoid function.

Similarly, given the embedding of the target legal case $\mathbf{e}(Y)$ and law article embeddings $\mathbf{e}(L_Y)$ and $\mathbf{e}(L_X)$, we can also get the reconstructed treatment through IV regression and recombination:

$$\begin{aligned} \widehat{\mathbf{e}}(Y) =& f^t_{\text{proj}}\left(c_t(L_Y, L_X)\right), \\ \widetilde{\mathbf{e}}(Y) =& \mathbf{e}(Y) - \widehat{\mathbf{e}}(Y), \\ \mathbf{e}^{\text{re}}(Y) =& \widehat{\mathbf{e}}(Y) + \alpha_t \cdot \widetilde{\mathbf{e}}(Y), \end{aligned} \tag{4}$$

where f^t_{proj}, c_t, and α_t are defined similarly as their counterparts (i.e., f^s_{proj} and c_s in Equation (1), and α_s in Equation (3)).

4.3 Model-agnostic application

Many document matching models share a similar structure, which we refer to as the underlying model. The underlying models represent each input document as an embedding vector and predict the matching score based on the representations. Law-Match is a model-agnostic framework implemented over existing document matching models that follow this underlying structure by feeding the reconstructed treatments to the matching model.

Formally, given a pair of legal cases (X, Y), the two treatment reconstruction modules respectively output the reconstructed embeddings $\mathbf{e}^{\text{re}}(X)$ and $\mathbf{e}^{\text{re}}(Y)$. Then, the matching score between X and Y can be calculated as:

$$\hat{z} = f_{\text{pred}}(\mathbf{e}^{\text{re}}(X), \mathbf{e}^{\text{re}}(Y)), \tag{5}$$

where f_{pred} can be any of the underlying models such as Sentence-BERT [25], Lawformer [41], Bert-PLI [29], IOT-Match [46] etc.

4.4 Model training

Law-Match has parameters to learn, including those in treatment reconstruction module for X (i.e., parameters in f^s_{proj}, f^s_{attn}, and f^s_{weight}) and those in treatment reconstruction module for Y. We denote these parameters as Θ_1. The underlying matching model f_{pred} also has another set of learn-able parameters, denoted as Θ_2. Law-Match designs an alternative optimization procedure for learning these parameters based on the labelled training data $\mathcal{D}$. Each optimization iteration consists of two stages: the IV regression stage for updating Θ_1 and the matching stage for updating Θ_2.

At each batch of the IV regression stage, after sampling n training pairs and the cited law articles, IV regression is employed to output the law-related representations $\widehat{\mathbf{e}}(X_i)$ and $\widehat{\mathbf{e}}(Y_i)$ for all $i = 1, \cdots, n$. MSE (mean square error) is used to measure the losses during the IV regression stage,

$$\mathcal{L}_{\text{IV}} = \frac{1}{n}\sum_{i=1}^{n}\left\{\|\widehat{\mathbf{e}}(X_i) - \mathbf{e}(X_i)\|^2 + \|\widehat{\mathbf{e}}(Y_i) - \mathbf{e}(Y_i)\|^2\right\}. \tag{6}$$

Gradients are then calculated to update the parameters in Θ_1.

Moving to the matching stage and at each batch, after sampling n training pairs and the cited law articles, the predicted matching scores $\hat{z}_i$'s are calculated according to current parameter values. Cross entropy is employed to measure the matching loss:

$$\mathcal{L}_{\text{match}} = \frac{1}{n}\sum_{i=1}^{n}\text{CE}(\widehat{z}_i, z_i), \tag{7}$$

where $\text{CE}(\widehat{z}_i, z_i)$ denotes the cross-entropy between the prediction $\widehat{z}_i$ and the ground-truth label z_i. Gradients are calculated to update the underlying matching model's parameters Θ_2.

5 DISCUSSION

In real-world applications, Law-Match may face several issues. The following subsections discuss how to address them.

5.1 Missing of the query law articles

In some real tasks such as legal retrieval, the law articles may not be provided in the source case X. One reason is that the source cases (queries) are not judged yet. To address the issue, we adapt the causal discovery method proposed in [18] to predict the missing law articles in the legal case[5].

The causal discovery method consists of two steps: (1) constructing a bipartite graph $\mathcal{G}$ for describing the relation between legal cases and law articles based on large-scale legal cases with cited law articles; and (2) inferring the law articles for the given legal cases based on $\mathcal{G}$.

The first step aims to create a bipartite graph $\mathcal{G} = (U, V, E)$ where U is the set of C sentence clusters, V is the set of M law article IDs, and E is the set of edges from V to U. The graph is created based on 100,000 legal cases with law articles, crawled from https://www.faxin.cn. Each case contains sentences describing the facts and a set of cited law articles IDs. Following the practices in the Section 3 of [18], $\mathcal{G}$ can be created by selecting key sentences from each legal case, clustering the key sentences, filtering unimportant sentences associated with the law articles, and finally creating links (E) that link the law article IDs (U) to clusters IDs (V). Please refer to [18] for the detailed procedure.

In the second step, given the legal case X and graph $\mathcal{G}$, the law articles can be predicted as follows:

(1) Split X into sentences $x_1, \cdots, x_K$;

(2) Assigning sentences x_i ($i = 1, \cdots, K$) to the clusters in U, using the embeddings generated by a pre-trained LMs and Euclidean distance. Note one sentence may be assigned to multiple clusters.

(3) For each cluster, selecting at most K' assigned sentences, according to the summed distances between sentences and U.

(4) For each x_i ($i = 1, \cdots, K$), collecting the set of associated law article IDs $\mathcal{A}_i$ (i.e., moving one step, starting from the assigned clusters (nodes in U) and following the edges in E);

(5) Return $\bigcup_{i=1}^{K}\mathcal{A}_i$.

[5]Note that causal discovery in [18] is originally designed for the task of similar charge disambiguation. Two modifications are made to adapt for predicting law articles: replacing the charge names with law articles and changing the operation level from the terms to sentences.

5.2 Underlying models that use paragraph inputs

Some legal case matching models, such as Bert-PLI [29] require paragraph embeddings, not the document embeddings, as inputs. For these models, Law-Match considers each paragraph of the legal case as a (pseudo) legal case. Therefore, each paragraph can be processed individually, achieving a single reconstructed vector. Let us use Law-Match, which uses BERT-PLI as the underlying model, as an example. Suppose that in (X, Y), X contains m paragraphs and Y contains n paragraphs. Law-Match will apply its treatment reconstruction $m \times n$ times. Each time, it takes a different paragraph pair (constructed based on X and Y) as the input, generating a pair of reconstructed paragraph embeddings. After that, these $m \times n$ embedding pairs are fed to BERT-PLI for conducting the final matching prediction.

Note that the paragraphs in a legal case have no law articles associated because the law articles are usually cited at the end of legal cases. Law-Match predicts the law articles for each paragraph using the causal discovery described in Section 5.1. Specifically, (1) if the original legal case also cites no law articles, for each paragraph in the case, we directly use the procedure in Section 5.1, and return the predicted law articles; (2) if the original legal case cites law articles, we first use the cited law articles to prune the graph $\mathcal{G}$, i.e., removing those nodes V (law articles IDs) that are not cited by the original legal case. Then, we use the procedure in Section 5.1 with the reduced graph to predict law articles for each paragraph.

5.3 Feasibility of using law articles as IVs

This section discusses whether law articles are valid IVs for legal case matching. A valid IV need to satisfy three assumptions: *Relevance*, *Exclusion Restriction* and *Instrumental Unconfoundedness* [24].

As for *Relevance*, it means that the IVs (law articles) need to be relevant to the treatments (legal cases). In order to verify the relevance, we use distance correlation (dCor) [35], which measures linear and nonlinear associations between two random variables, to measure the relevance between treatments and IVs. dCor$\in [0, 1]$ where a larger dCor means more relevant. For each legal case in ELAM (details Section 6.1.1), we calculate the dCors value between the cited law articles and the fact descriptions of the legal case. The averaged dCor= 0.7327. As for comparison, we replace the cited law articles with law articles randomly selected from a law book. The averaged dCor decreases to dCor= 0.2673. The result indicates that the cited law articles satisfy the relevance assumption.

As for *Exclusion Restriction*, the IVs causal effect on the outcome is fully mediated by the treatment. Law-Match estimates the similarities between the source and target cases. The law articles are associated to concrete legal cases rather than the matching labels. This means the law articles can only effect the matching labels through concrete legal cases. Therefore, Law-Match meets the *Exclusion Restriction* assumption.

As for *Instrumental Unconfoundedness*, the IVs need to be uncorrelated with the confounders. In Law-Match, the confounders could be some missing variables (e.g., the focus of disputes) unrelated to any law articles. The law articles only affect the key elements in the legal case. Therefore, the law articles are independent of the confounder, satisfying the exogeneity assumption.

6 EXPERIMENTS

In this section, we empirically verify the efficiency of Law-Match. The source code and all experiments have been shared at https://github.com/Jeryi-Sun/Law-Match-SIGIR-23.

6.1 Experimental settings

6.1.1 Datasets. The experiments were conducted based on three publicly available datasets: ELAM [46], eCAIL [46], and LeCaRD [19].

ELAM is an explainable legal case matching dataset. It contains 1250 source legal cases, each associated with four target cases. Each legal case pair is manually assigned a matching label which is either match (2), partially match (1), or mismatch (0). Explainable labels such as rationales, their alignments, and free-form explanations are also provided in the dataset. In the experiments, we use the legal case contents and the matching labels for training and evaluating the matching models.

eCAIL is an extension of CAIL (Challenge of AI in Law) 2021 dataset[6]. In CAIL data, each legal case is associated with tags about private lending. Following the practices in [46], we constructed 1875 source cases, each associated with four target cases. Each legal case pair is assigned a matching label according to the number of overlapping tags (match if overlapping > 10, mismatch if < 1, and partially match otherwise).

LeCaRD is a legal case retrieval dataset which contains 107 source (query) cases and 43,000 target cases. All criminal cases were published by the Supreme People's Court of China. For each query, 30 target cases are manually annotated, each assigned a 4-level relevance (matching) label.

As for the law articles, we count each dataset's occurrences of different law categories. Specifically, for LeCaRD and ELAM, we use the PRC Criminal Law; for eCAIL, we use the PRC Contract Law and Civil Procedure Law. The contents of law articles are downloaded from https://flk.npc.gov.cn/.

6.1.2 Baselines and evaluation metrics. The proposed Law-Match is a model-agnostic framework, which is applied to the following underlying models:

Sentence-BERT [25] is a text matching model. It uses BERT [10] to encode two sentences separately. Then it concatenates the two embeddings together and uses an MLP to conduct matching.

Lawformer [41] is a Longformer-based pre-trained language model training millions of Chinese legal cases to represent long legal documents better. In the experiment, we send the texts of two cases together to Lawformer and use the mean pooling of Lawformer's output to conduct matching.

BERT-PLI [29] uses BERT to capture the semantic relationships at the paragraph level. Then it uses RNN and Attention model to infer the relevance between the two cases. Finally, it uses an MLP to calculate the aggregated embeddings similarity score.

IOT-Match [46] is a three-stage model designed for explainable legal case matching. It extracts rationales in the first stage, generates natural language-based explanations in the second stage, and conducts explainable legal case matching in the third stage. We kept the first two stages identical to [46] and conducted Law-Match at the third stage. Note that IOT-Match needs explainable

[6] Fact Prediction Track data: http://cail.cipsc.org.cn/

features which are unavailable in LeCaRD, we only compared with IOT-Match on ELAM and eCAIL.

We also compare Law-Match with two baselines that can also add the law articles' knowledge to legal cases. The first is an intuitive baseline that simply appends the contents of cited law articles to the original cases, forming new extended legal cases. Existing matching models of Sentence-BERT, Lawformer, BERT-PLI and IOT-Match can be applied to the extended legal cases, denoted as **Cat-Law (Sentence-BERT)**, **Cat-Law (Lawformer)**, **Cat-Law (BERT-PLI)**, and **Cat-Law (IOT-Match)**, respectively. The second is from [14] which employs an attention mechanism to incorporate article semantics into the legal judgement prediction models called EPM. Existing matching models of Sentence-BERT, Lawformer, BERT-PLI and IOT-Match can be applied to EPM, denoted as **EPM (Sentence-BERT)**, **EPM (Lawformer)**, **EPM (BERT-PLI)**, and **EPM (IOT-Match)**, respectively.

The proposed Law-Match is model-agnostic. In the experiments, we applied Law-Match to the baselines of Sentence-BERT, Lawformer, BERT-PLI, and IOT-Match achieving four versions, referred to as **Law-Match (Sentence-BERT)**, **Law-Match (Lawformer)**, **Law-Match (BERT-PLI)**, and **Law-Match (IOT-Match)** respectively.

As for evaluation metrics, we use Accuracy, Macro-Precision, Macro-Recall, and Macro-F1 to measure the matching accuracy.

6.1.3 Implementation details. Law-Match's hyperparameters are tuned using grid search on the validation set with Adam [16]. The batch size is tuned among $\{2, 4, 8\}$. The learning rate η_1 and η_2 are tuned among $\{3e-6, 3e-5, 3e-4\}$. For the source cases that do not cite law articles, the number of automatically discovered law articles K' is tuned between [3, 15] with step 2. For baselines, we set the parameters as the optimal values in the original paper.

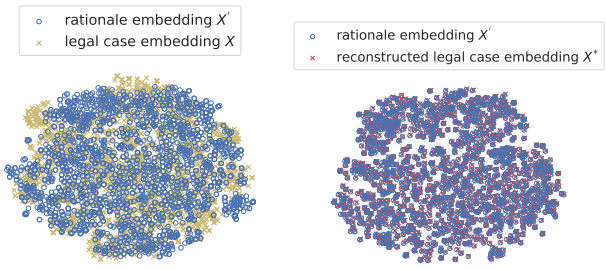
We use Legal-Bert[7] to encode legal cases (and the corresponding law articles if needed) for Sentence-BERT [25], Bert-PLI [29], and IOT-Match [46]. The legal cases from eCAIL are generally longer than BERT's maximum input length. For Sentence-BERT [25], we use TextRank [20] to process the legal cases and generate a summary with a 512-words for each case. For Bert-PLI [29] and Lawformer [41], the original text is used.

6.2 Experimental results and analysis

6.2.1 ***Comparison against underlying models and baselines.*** From the results reported in Table 1, we found that Law-Match (Sentence-BERT), Law-Match (Lawformer), and Law-Match (BERT-PLI) outperformed the corresponding underlying models (i.e., Sentence-BERT, Lawformer, and BERT-PLI) on all of the three datasets (ELAM, eCAIL, and LeCaRD) and Law-Match (IOT-Match) outperformed IOT-Match on ELAM and eCAIL, with statistical significance (t-tests, p-value <0.05). The results verified the efficiency of the model-agnostic Law-Match framework in improving the underlying matching models.

Meanwhile, we find that the four versions of Cat-Law/EPM, i.e., Cat-Law/EPM (Sentence-BERT), Cat-Law/EPM (Lawformer), Cat-Law/EPM (BERT-PLI), and Cat-Law/EPM (IOT-Match) also outperform most of the underlying models, indicating that the knowledge from the law articles helps improve legal case matching. Finally, Law-Match (Sentence-BERT), Law-Match (Lawformer), Law-Match (BERT-PLI), and Law-Match(IOT-Match) outperform the corresponding baselines of Cat-Law/EPM (Sentence-BERT), Cat-Law/EPM (Lawformer), Cat-Law/EPM (BERT-PLI), and Cat-Law/EPM (IOT-Match), verify that considering law articles as IVs to decompose treatments is a better way of using law articles in legal cases matching.

[7] https://github.com/thunlp/OpenCLaP



(a) average distance: 13.20 (b) average distance: 0.28

Figure 4: (a): Embeddings of legal case generated by Legal-Bert v. Embeddings of rationale generated by Legal-Bert. (b): Embeddings of legal case generated by Legal-Bert v. Reconstructed Embeddings of legal case generated by Law-Match.

6.2.2 ***How the law articles improve legal case matching?*** We first show that Law-Match has the ability of *Identifying Rationales*. That is, the law articles guide Law-Match to reconstruct the legal case embeddings that focus more on the rationales, which have been verified to be beneficial to accurate matching [46]. Specifically, we note that each legal case in ELAM also contains human-annotated rationales (key sentences). Therefore, for each legal case, we generate the legal case embedding X by Legal-Bert, the rationale embedding X' which contains only the human-annotated rationales by Legal-Bert, and the reconstructed legal cases embedding X^* by Law-Match.

First, we use TSNE [36] to illustrate all of the 2500 legal case embeddings in Figure 4. Figure 4(a) shows the distributions of the legal case embeddings X (yellow crosses) and the rationale embeddings X' (blue circles). Figure 4(b) shows the distributions of the reconstructed legal case embeddings X^* (red crosses) and the rationale embeddings X' (blue circles). It is easy to observe that the blue circles and yellow crosses in Figure 4(a) are distributed more differently than the blue circles and red crosses in Figure 4(b). That is, more circles and crosses are not overlapped in Figure 4(a). Moreover, we calculate the averaged Euclidean distances over all of the pairs (X, X') in Figure 4(a) based on embedding vectors generated by Legal-Bert. The average distance is 13.20. After applying Law-Match on the legal cases and based on the reconstructed embeddings, the average Euclidean distances over all of the pairs (X^*, X') in Figure 4(b) becomes 0.28. The results verify that by using law articles as IVs, Law-Match reconstruct the case embeddings so that they are closer to the corresponding rationale embeddings.

We further show that Law-Match has the ability of *disentangling the law-related and law-unrelated parts* of treatment vectors

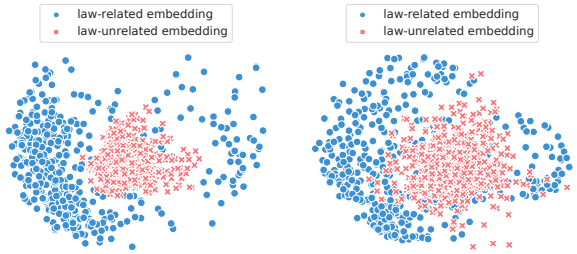
Table 1: Performance comparisons between Law-Match and the baselines. The boldface represents the best performance. In each block, we present the Law-Match in the last line. '†' indicates the improvements over all of the baselines are statistically significant (t-tests, p-value < 0.05). Results of IOT-Match on LeCaRD are not available, denoted as '—'.

Models	ELAM				LeCaRD				eCAIL			
	Acc. (%)	P. (%)	R. (%)	F1 (%)	Acc. (%)	P. (%)	R. (%)	F1 (%)	Acc. (%)	P. (%)	R. (%)	F1 (%)
Sentence-BERT	68.83	69.83	66.88	67.20	59.44	59.54	57.89	58.70	71.33	70.83	71.21	70.98
Cat-Law(Sentence-BERT)	71.54	70.54	69.73	69.94	61.60	62.54	59.76	60.73	78.80	78.36	78.70	78.53
EPM(Sentence-BERT)	71.14	69.85	69.51	69.65	60.37	61.56	58.12	59.25	77.06	76.65	76.95	76.58
Law-Match(Sentence-BERT)	**73.15**†	**71.23**†	**71.05**†	**71.14**†	**62.54**†	**63.37**†	**61.04**†	**61.84**†	**80.00**†	**79.78**†	**79.92**†	**79.84**†
Lawformer	69.91	72.26	68.34	69.18	59.13	58.79	58.56	58.47	70.67	70.20	70.55	69.91
Cat-Law(Lawformer)	69.94	68.05	68.40	68.22	59.13	59.27	58.54	58.59	75.19	75.51	75.18	75.20
EPM(Lawformer)	71.14	72.75	70.58	70.31	59.37	60.02	59.14	59.43	74.00	73.85	74.20	74.00
Law-Match(Lawformer)	**73.20**†	**74.41**†	**73.12**†	**73.52**†	**60.06**†	**60.80**†	**59.54**†	**59.62**†	**76.67**†	**76.25**†	**76.56**†	**76.40**†
BERT-PLI	71.21	71.22	71.23	70.88	61.60	60.88	60.41	60.48	70.66	70.05	70.54	70.18
Cat-Law(BERT-PLI)	72.89	71.32	70.49	70.63	63.46	64.15	62.47	63.16	73.20	72.51	73.08	72.28
EPM(BERT-PLI)	71.34	69.32	69.11	68.99	63.77	65.26	62.45	63.49	73.33	73.12	73.23	73.18
Law-Match(BERT-PLI)	**74.95**†	**72.96**†	**71.75**†	**72.35**†	**65.63**†	**66.07**†	**63.75**†	**64.41**†	**74.13**†	**73.51**†	**74.02**†	**73.68**†
IOT-Match	73.87	73.02	72.41	72.55	—	—	—	—	82.01	82.10	81.92	81.90
Cat-Law(IOT-Match)	74.55	73.22	72.63	72.89	—	—	—	—	83.86	84.59	83.72	83.95
EPM(IOT-Match)	74.69	73.39	73.02	73.17	—	—	—	—	82.53	82.21	82.40	82.29
Law-Match(IOT-Match)	**76.75**†	**75.51**†	**75.78**†	**75.59**†	—	—	—	—	**84.60**†	**84.44**†	**84.53**†	**84.45**†

Table 2: Ablation study of Law-Match on ELAM.

Algorithm	Sentence-BERT				Lawformer				Bert-PLI				IOT-Match			
	Acc. (%)	P. (%)	R. (%)	F1 (%)	Acc. (%)	P. (%)	R. (%)	F1 (%)	Acc. (%)	P. (%)	R. (%)	F1 (%)	Acc. (%)	P. (%)	R. (%)	F1 (%)
Law-Match (fitted only)	66.53	66.47	64.69	65.15	71.74	69.90	69.33	69.35	70.74	68.71	68.35	68.52	70.74	69.89	69.31	69.45
Law-Match (residual only)	69.13	68.01	68.19	68.10	71.14	69.37	68.68	68.70	69.53	67.05	66.19	66.62	73.34	71.83	72.33	71.96
Law-Match (Concat parts)	71.74	70.55	70.65	70.54	71.03	70.87	70.56	70.34	73.55	71.52	71.68	71.52	74.95	74.11	74.31	74.20
Law-Match (Separate IV)	72.55	71.05	70.73	70.82	71.14	71.00	69.78	70.12	71.74	69.15	68.72	68.93	74.54	73.14	69.63	69.30
Law-Match	**73.15**	**71.23**	**71.05**	**71.14**	**73.20**	**74.41**	**73.12**	**73.52**	**74.95**	**72.96**	**71.75**	**72.35**	**76.75**	**75.51**	**75.78**	**75.59**

in Section 4. Specifically, we visualize the decomposed law-related and law-unrelated embeddings learned by Law-Match using TSNE. Figure 5(a) and (b) show the results w.r.t. Law-Match(Sentence-BERT) and Law-Match(Lawformer), respectively. The law-related embeddings are shown as blue dots, and the law-unrelated embeddings are shown as red crosses. In Figure 5, we observe that Law-Match separates the two sets of embeddings. Only a tiny fraction of the vectors are overlapped. From the above analysis, we conclude that Law-Match effectively disentangles the law-related and law-unrelated parts of treatment vectors, which are utilized differently and enhanced legal case matching.



(a) Law-Match(Sentence-BERT) (b) Law-Match(Lawformer)

Figure 5: Visualization of the learned causal and non-causal embeddings of (a) Law-Match(Sentence-BERT) and (b) Law-Match(Lawformer). Causal parts are represented by dots and non-causal parts are represented by crosses. Using law articles as IVs, causal and non-causal parts are disentangled clearly by Law-Match.

6.2.3 ***Ablation study.*** Law-Match combines the fitted and residual parts as the reconstructed legal case representation. We create several Law-Match variations by removing the two parts or changing the combination methods. They are (a) Law-Match (fitted only): only use the fitted part as the reconstructed representation; (b) Law-Match (residual only): only use the residual part as the reconstructed representation. Also, note that in Law-Match, the embeddings of L_X and L_Y are used as IVs to regress both X and Y. We created another variation: (c) Law-Match (Separate IV): L_X is only used as the IV for X, and L_Y is only used as the IV for Y; (d) Law-Match (Concat parts): simply concatenate the fitted part and the residual part as the reconstructed representation.

Table 2 reports the performance of these variations with different underlying models on the ELAM dataset. The results indicate that (1) removing either the fitted or residual part will decrease the matching performance; (2) combining L_X and L_Y's embeddings

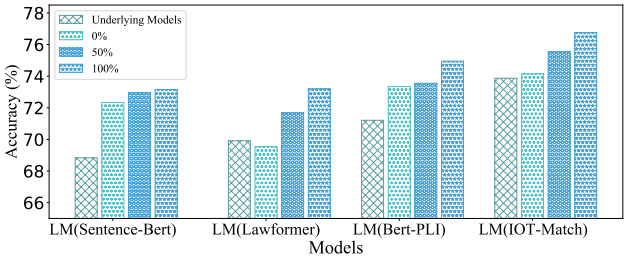


Figure 6: Performance of Law-Match (abbr. LM) w.r.t. different ratio of (oracle) cited law articles on ELAM.

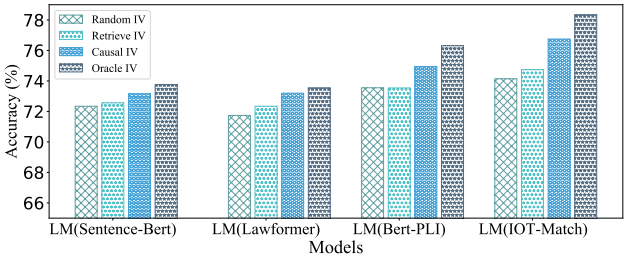


Figure 7: Performance of Law-Match (abbr. LM) w.r.t. law articles discovered by different methods on ELAM.

as IVs can further enhance the matching accuracy; (3) simply concatenating the fitted part and the residual part will decrease the matching performance.

6.2.4 ***Robustness of law articles as IVs.*** Based on the ELAM dataset, we test the performances of Law-Match when a few oracle law articles (those cited in the legal cases) are replaced with those randomly selected from a law book. That is, we try to inject noise into the IVs. Figure 6 illustrates the matching accuracy of Law-Match w.r.t. 0%, 50%, and 100% of the oracle law articles are kept (others are replaced with random law articles) and the underlying modes without Law-Match. The results indicate that: (1) Law-Match is robust. It improved the underlying matching models even the law articles are randomly selected; (2) high-quality law articles can further enhance the matching accuracy.

6.2.5 ***Effects of the causally discovered law articles.*** We test the efficiency of automatically discovered law articles (shown in Section 5.1) in Law-Match. Specifically, we note that the original legal cases in ELAM contain oracle-cited law articles. We compare the matching accuracy of Law-Match variations where the law articles are collected differently: (1) randomly selecting 5 law articles[8], denoted as Law-Match (Random IV); (2) using BM25 [26] to retrieve top-5 law articles from the law book, where legal cases and law articles are respectively considered as queries and documents, denoted as Law-Match (Retrieve IV); (3) using the causal discovery method presented in subsection 5.1, denoted as Law-Match (Causal IV); (4) For showing the upper bound performance, we also test Law-Match with oracle cited law articles in the legal cases, denoted as Law-Match (Oracle IV).

[8]On average each ELAM legal case cites about 5 law articles.

Figure 7 shows that Law-Match (Causal IV) perform better than Law-Match (Retrieve IV) and Law-Match (Retrieve IV) perform better than Law-Match (Random IV). Law-Match (Oracle IV) perform the best. The results verify the efficiency of the causal discovery module presented in subsection 5.1, especially when no law articles are cited in the legal cases.

Table 3: Average online inference time per case pair, for four base models w/ or w/o Law-Math.

Model	Sentence-Bert	Lawformer	Bert-PLI	IOT-Match
w/o Law-Match	0.0502 (s)	0.0318 (s)	0.1228 (s)	0.1027 (s)
w/ Law-Match	0.0543 (s)	0.0341 (s)	0.1269 (s)	0.1059 (s)
RelaCost	8.17 %	7.23 %	3.34 %	3.12 %

6.2.6 ***Efficiency of Law-Match.*** We also analyze the efficiency of Law-Match to show the additional time needed when conducting online matching. Specifically, we record the time required to process each case pair in the online inference stage, with different base models with and without the Law-Match module. From the results reported in Table 3, we find that Law-Match needs a short additional time when applied to different base models. Also, we find that relative additional time costs are lower for the larger base models. Overall, the delay is acceptable and will not impact the online inference speed much.

The results are reasonable because Law-Match is designed as an independent representation learning module in the process of legal matching. The most time-consuming part is reconstructing the treatment, which consists of four MLP layers ($f^{s/t}_{\text{weight}}$, $f^{s/t}_{\text{proj}}$) and two additive attention networks ($f^{s/t}_{\text{attn}}$). These modules require much less time than the base models.

7 CONCLUSIONS

In this paper, we propose a model-agnostic causal learning framework that introduces law articles to legal case matching, called Law-Match. Analyses show that the legal case matching results are affected by the mediation effect of the cited law articles and the direct effect of the key circumstances in legal cases. By considering the law articles as IVs and legal cases as treatments, Law-Match uses IV regression to decompose each legal case's embedding into the law-related and law-unrelated parts, which are then combined together for the final matching prediction. Experiments on three public datasets demonstrated the efficiency of Law-Match.

ACKNOWLEDGMENTS

This work was funded by the National Key R&D Program of China (2019YFE0198200), Beijing Outstanding Young Scientist Program NO. BJJWZYJH012019100020098, Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China (23XNH024), Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, and Public Computing Cloud, Renmin University of China. The work was partially done at Beijing Key Laboratory of Big Data Management and Analysis Methods.

REFERENCES

[1] Joshua D Angrist and Guido W Imbens. 1995. Two-stage least squares estimation of average causal effects in models with variable treatment intensity. *Journal of the American statistical Association* 90, 430 (1995), 431–442.

[2] Joshua D Angrist, Guido W Imbens, and Donald B Rubin. 1996. Identification of causal effects using instrumental variables. *Journal of the American statistical Association* 91, 434 (1996), 444–455.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[4] Andrea Bellavia and Linda Valeri. 2018. Decomposition of the total effect in the presence of multiple mediators and interactions. *American journal of epidemiology* 187, 6 (2018), 1311–1318.

[5] Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150* (2020).

[6] T. Bench-Capon, Micha Araszkiewicz, K. Ashley, K. Atkinson, F. Bex, F. Borges, D. Bourcier, P. Bourgine, J. G. Conrad, and E. Francesconi. 2012. A history of AI and Law in 50 papers: 25 years of the international conference on AI and Law. *Artificial Intelligence & Law* 20, 3 (2012), 215–319.

[7] 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*. 1657–1660.

[8] Paheli Bhattacharya, Kripabandhu Ghosh, Arindam Pal, and Saptarshi Ghosh. 2020. Methods for computing legal document similarity: A comparative study. *arXiv preprint arXiv:2004.12307* (2020).

[9] Mehmet Caner and Bruce E Hansen. 2004. Instrumental variable estimation of a threshold model. *Econometric Theory* 20, 5 (2004), 813–843.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT (1)*. Association for Computational Linguistics, 4171–4186.

[11] João Dias, Pedro A Santos, Nuno Cordeiro, Ana Antunes, Bruno Martins, Jorge Baptista, and Carlos Gonçalves. 2022. State of the Art in Artificial Intelligence applied to the Legal Domain. *arXiv preprint arXiv:2204.07047* (2022).

[12] Christian Dippel, Andreas Ferrara, and Stephan Heblich. 2020. Causal mediation analysis in instrumental-variables regressions. *The Stata Journal* 20, 3 (2020), 613–626.

[13] Rahul M Dodhia. 2005. A review of applied multiple regression/correlation analysis for the behavioral sciences. *Journal of Educational and Behavioral Statistics* 30, 2 (2005), 227–229.

[14] 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)*. Association for Computational Linguistics, Dublin, Ireland, 648–664. https://doi.org/10.18653/v1/2022.acl-long.48

[15] Jason Hartford, Greg Lewis, Kevin Leyton-Brown, and Matt Taddy. 2017. Deep IV: A flexible approach for counterfactual prediction. In *International Conference on Machine Learning*. PMLR, 1414–1423.

[16] Diederik P Kingma and Jimmy Ba Adam. 2015. A Method for Stochastic. *Optimization. In, ICLR* 5 (2015).

[17] Sushanta Kumar, P Krishna Reddy, V Balakista Reddy, and Aditya Singh. 2011. Similarity analysis of legal judgments. In *Proceedings of the Fourth Annual ACM Bangalore Conference*. 1–4.

[18] Xiao Liu, Da Yin, Yansong Feng, Yuting Wu, and Dongyan Zhao. [n. d.]. Everything Has a Cause: Leveraging Causal Inference in Legal Text Analysis. ([n. d.]).

[19] Yixiao Ma, Yunqiu Shao, Yueyue Wu, Yiqun Liu, Ruizhe Zhang, Min Zhang, and Shaoping Ma. 2021. LeCaRD: A Legal Case Retrieval Dataset for Chinese Law System. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2342–2348.

[20] Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*. 404–411.

[21] Akshay Minocha, Navjyoti Singh, and Arjit Srivastava. 2015. Finding relevant indian judgments using dispersion of citation network. In *Proceedings of the 24th International Conference on World Wide Web*. 1085–1088.

[22] Ziang Niu, Yuwen Gu, and Wei Li. 2022. Estimation and inference for high-dimensional nonparametric additive instrumental-variables regression. *arXiv e-prints* (2022), arXiv–2204.

[23] Judea Pearl. 2009. *Causality*. Cambridge university press.

[24] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. 2017. *Elements of causal inference: foundations and learning algorithms*. The MIT Press.

[25] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. 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)*. 3982–3992.

[26] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389.

[27] Manavalan Saravanan, Balaraman Ravindran, and Shivani Raman. 2009. Improving legal information retrieval using an ontological framework. *Artificial Intelligence and Law* 17, 2 (2009), 101–124.

[28] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. 2021. Toward causal representation learning. *Proc. IEEE* 109, 5 (2021), 612–634.

[29] Yunqiu Shao, Jiaxin Mao, Yiqun Liu, Weizhi Ma, Ken Satoh, Min Zhang, and Shaoping Ma. 2020. BERT-PLI: Modeling Paragraph-Level Interactions for Legal Case Retrieval.. In *IJCAI*. 3501–3507.

[30] J Myles Shaver. 2005. Testing for mediating variables in management research: Concerns, implications, and alternative strategies. *Journal of management* 31, 3 (2005), 330–353.

[31] Zihua Si, Xueran Han, Xiao Zhang, Jun Xu, Yue Yin, Yang Song, and Ji-Rong Wen. 2022. A Model-Agnostic Causal Learning Framework for Recommendation using Search Data. In *Proceedings of the ACM Web Conference 2022*. 224–233.

[32] Zihua Si, Zhongxiang Sun, Xiao Zhang, Jun Xu, Yang Song, Xiaoxue Zang, and Ji-Rong Wen. 2023. Enhancing Recommendation with Search Data in a Causal Learning Manner. 41, 4 (2023).

[33] James H Stock and Francesco Trebbi. 2003. Retrospectives: Who invented instrumental variable regression? *Journal of Economic Perspectives* 17, 3 (2003), 177–194.

[34] Zhongxiang Sun. 2023. A Short Survey of Viewing Large Language Models in Legal Aspect. *arXiv preprint arXiv:2303.09136* (2023).

[35] Gábor J Székely and Maria L Rizzo. 2014. Partial distance correlation with methods for dissimilarities. *The Annals of Statistics* 42, 6 (2014), 2382–2412.

[36] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[37] Tyler J VanderWeele. 2013. A three-way decomposition of a total effect into direct, indirect, and interactive effects. *Epidemiology (Cambridge, Mass.)* 24, 2 (2013), 224.

[38] Arun Venkatraman, Wen Sun, Martial Hebert, J Bagnell, and Byron Boots. 2016. Online instrumental variable regression with applications to online linear system identification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 30.

[39] Jeffrey M Wooldridge. 2015. *Introductory econometrics: A modern approach*. Cengage learning.

[40] Anpeng Wu, Kun Kuang, Bo Li, and Fei Wu. 2022. Instrumental variable regression with confounder balancing. In *International Conference on Machine Learning*. PMLR, 24056–24075.

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

[42] Liyuan Xu, Yutian Chen, Siddarth Srinivasan, Nando de Freitas, Arnaud Doucet, and Arthur Gretton. 2020. Learning Deep Features in Instrumental Variable Regression. In *International Conference on Learning Representations*.

[43] 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*. 3086–3095.

[44] Mengyue Yang, Furui Liu, Zhitang Chen, Xinwei Shen, Jianye Hao, and Jun Wang. 2021. CausalVAE: Disentangled representation learning via neural structural causal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9593–9602.

[45] Weijie Yu, Liang Pang, Jun Xu, Bing Su, Zhenhua Dong, and Ji-Rong Wen. 2022. Optimal Partial Transport Based Sentence Selection for Long-form Document Matching. In *Proceedings of the 29th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Gyeongju, Republic of Korea, 2363–2373. https://aclanthology.org/2022.coling-1.208

[46] Weijie Yu, Zhongxiang Sun, Jun Xu, Zhenhua Dong, Xu Chen, Hongteng Xu, and Ji-Rong Wen. 2022. Explainable Legal Case Matching via Inverse Optimal Transport-based Rationale Extraction. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 657–668.

[47] Junkun Yuan, Anpeng Wu, Kun Kuang, Bo Li, Runze Wu, Fei Wu, and Lanfen Lin. 2022. Auto IV: Counterfactual Prediction via Automatic Instrumental Variable Decomposition. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 16, 4 (2022), 1–20.

[48] Yiming Zeng, Ruili Wang, John Zeleznikow, and Elizabeth Kemp. 2005. Knowledge representation for the intelligent legal case retrieval. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*. Springer, 339–345.

[49] Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Chaojun Xiao, Zhiyuan Liu, and Maosong Sun. 2018. Legal judgment prediction via topological learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 3540–3549.