Legal Judgment Prediction: A Survey of the State of the Art

Yi Feng[1] , **Chuanyi Li**[1] and **Vincent Ng**[2]
[1]State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
[2]Human Language Technology Research Institute, University of Texas at Dallas, Richardson, Texas, USA
fy@smail.nju.edu.cn, lcy@nju.edu.cn, vince@hlt.utdallas.edu

Abstract

Automatic legal judgment prediction (LJP) has recently received increasing attention in the natural language processing community in part because of its practical values as well as the associated research challenges. We present an overview of the major milestones made in LJP research covering multiple jurisdictions and multiple languages, and conclude with promising future research directions.

1 Introduction

Legal judgment prediction (LJP) refers to a collection of tasks that involve predicting the court's outcome given the facts of a legal case [Aletras *et al.*, 2016; Chalkidis *et al.*, 2019; Feng *et al.*, 2021] and possibly other information in the case description such as the arguments and the claims [Sulea *et al.*, 2017b; Malik *et al.*, 2021; Gan *et al.*, 2021]. Automatic LJP has practical significance: LJP systems may assist legal professionals in analyzing cases and provide consulting services to laymen while reducing legal costs and improving access to justice.

From a research perspective, one of the most interesting aspects of LJP is that it encompasses a set of tasks that vary in difficulty. Table 1 enumerates the six key LJP subtasks, the associated subtask descriptions, as well as the language(s) in which the corresponding subtask has been explored. Note that the six subtasks are listed in increasing order of difficulty. As can be seen, Violation and Court Decision are generally the easier subtasks as the number of labels involved in these classification tasks are fairly small. For instance, Violation is a 2-class classification task that involves predicting whether a law article is violated. Though being the easiest subtask, Violation is by no means trivial, as the system needs to thoroughly comprehend the details in the facts (e.g., identifying the amount involved in a theft, as a low financial value is not a crime in some jurisdictions). Further down the list, we see Charge and Law Article, which are comparatively more difficult as they are *multi-label* classification tasks in which the system needs to pinpoint the set of charges and law articles that are relevant to a given case. Being the most difficult LJP subtask, Prison Term prediction involves more sophisticated logic. More specifically, if a criminal is accused of several crimes, the total prison term is not simply the sum of each crime's prison term, and in addition, specific judgment rules

LJP Subtask	Description	Language
Violation	Predicting violation or not of any law article of a case	English Chinese
Court Decision	Predicting the court decision of a case, such as the approval/dismissal, the court ruling (e.g. reject, cassation, non-lieu etc.), the affirm/reverse	French English Italian German Chinese
Charge	Identifying the crime charges of which a defendant should be accused	Chinese
Law Article	Identifying the specific violated law articles for a case	English Chinese
Alleged Law Article	Predicting the alleged law articles (put forward by the plaintiffs) of a case	English
Prison Term	Estimating the prison term	Chinese

Table 1: Different subtasks of LJP.

(e.g., the total prison term should be less than the sum but more than the maximum of each single prison term, and surrender may decrease the total) need to be considered.

An important issue that is relevant to LJP is *explainability*: can a system provide a justification for its own decision that can be understood by law professionals or even laymen? The justification for a model's decision is essential in that it can increase a user's trust of the system given the gravity that legal outcomes have for individuals. Given the importance of explainability in LJP, we will examine existing approaches to explainability alongside approaches to various LJP subtasks.

Our goal in this paper is to provide the AI audience with an overview of the major milestones in LJP research. To our knowledge, there are no surveys that are written specifically on automatic LJP. The most relevant survey has the broader goal of providing a general overview of the state-of-the-art (SOTA) of AI in the legal domain, and discusses a variety of tasks such as Similar Case Matching, Legal Question Answering, and Legal Judgment Prediction) [Zhong *et al.*, 2020b]. Importantly, it by no means focuses on LJP. We therefore believe this timely survey will provide up-to-date knowledge of LJP and be of interest to AI researchers.

2 Corpora

In this section, we present six corpora that have been widely used for training and evaluating LJP systems, namely ECHR2019 [Chalkidis *et al.*, 2019], ECHR2021 [Chalkidis *et*

Corpus	Language	Jurisdiction	No. of Cases	Annotated LJP Subtasks (No. of Labels w.r.t. Subtask)	Additional Annotation
ECHR2019	English	Europe	11,478	Violation (2) Law Article (66)	the case importance
ECHR2021	English	Europe	11,000	Alleged Law Article (40) Violation (2) Law Article (40)	the paragraph-level rationale
CAIL2018	Chinese	China	2,676,075	Law Article (183) Charge (202) Prison Term (integer value)	the defendant the penalty of money
SJP	German French Italian	Switzerland	49,883 (German) 31,094 (French) 4,292 (Italian)	Court Decision (2)	the publication year the legal area the canton of origin
ILDC	English	India	34,816	Court Decision (2)	the sentence-level explanation
FCCR	French	France	126,865	Court Decision (6 and 8 w.r.t two setups)	the date of the court ruling the law area

Table 2: Comparison of several popularly used corpora for legal judgment prediction.

al., 2021], CAIL2018 [Xiao *et al.*, 2018], SJP [Niklaus *et al.*, 2021], ILDC [Malik *et al.*, 2021], and FCCR [Sulea *et al.*, 2017b]. Table 2 compares these corpora along five dimensions: (1) the language, (2) the jurisdiction, (3) the number of cases, (4) the subtasks of LJP annotated in the corpus and the size of the corresponding label inventory, and (5) additional annotations on the corpus.

A few points deserve mention. First, each corpus covers a subset of the LJP subtasks described in Table 1. For instance, Law Article is covered only by ECHR2019, ECHR2021 and CAIL2018; and Prison Term is only covered by CAIL2018. Second, some of the additional annotations are interesting in that they can enable LJP-related tasks to be studied. For instance, ECHR2021 provides paragraph-level rationales for each of the alleged article violations of a case. Specifically, these rationales are selected from the paragraphs that appear in the facts of the given case and can be used to study explainability issues. As another example, each case in SJP is annotated with publication years, legal areas and cantons of origin. These annotations can be used to examine issues related to fairness and robustness in LJP (e.g., does the canton of origin have any impact on the court's decision or a LJP system's prediction?). Finally, researchers have constructed two datasets out of CAIL2018: *CAIL-small* consists of 154592, 17131 and 32508 training, validation and test samples, respectively, whereas *CAIL-big* consists of 1710856 training samples and 217016 test samples.

Note that several not so commonly used LJP datasets in other jurisdictions or languages exist, such as the Supreme Court Database (an English dataset from the U.S.) [Katz *et al.*, 2017] and the German Legal Decision Corpora (a German dataset from Germany) [Urchs *et al.*, 2021].

3 Evaluation Metrics

Next, we discuss the metrics used to evaluate LJP systems.

As mentioned in the introduction, except for Prison Term prediction, all of the LJP subtasks are essentially classification tasks. While Prison Term is most naturally cast as a regression task (because the prison term is a real value), this LJP subtask is typically tackled as a multi-class classification task where the prison term is divided into non-overlapping intervals and a label is created for each interval for prediction purposes [Xu *et al.*, 2020].

Since virtually all LJP subtasks are classification tasks, researchers have used precision (P), recall (P), and F1 score as the primary evaluation metrics. Because the class distribution in some of the evaluation corpora is highly skewed, it is somewhat unfair to measure performance using micro-P, micro-R, and micro-F1 as micro-averaging gives more importance to labels having a higher representation [Xiao *et al.*, 2021]. As a result, it is more common to report system performance using macro-P, macro-R, and macro-F1 [Chalkidis *et al.*, 2021; Feng *et al.*, 2019; Feng *et al.*, 2021], which give the same importance to each label. Note that when Prison Term prediction is cast as a regression task where the number of months in prison is to be predicted, Mean Squared Error (MSE) and Exact Match (EM) have been used as evaluation metrics [Chen *et al.*, 2019; Wu *et al.*, 2020a].

4 Systems

In this section, we examine the key ideas involved in the development of existing LJP systems. While early work on LJP has employed rule-based approaches [Kort, 1957; Nagel, 1963; Segal, 1984], virtually all recent work has adopted learning-based approaches. To date, the most successful learning-based LJP systems are *supervised*, and hence the discussion below will be focused on supervised approaches. Nevertheless, we note that for the prediction of law articles, alleged law articles, and charges, the training sets of some of the corpora for these LJP subtasks have skewed class distributions, where some of the class labels (i.e., law articles or charges) appear infrequently. Hence, some researchers have handled these subtasks in a *few-shot* classification setting [Hu *et al.*, 2018; Chalkidis *et al.*, 2019].

As noted before, with the exception of Prison Term prediction, which can be cast as a classification or regression task, the rest of the LJP subtasks are being cast as classification tasks. As a result, virtually all supervised LJP systems are classification-based. Broadly, supervised approaches to LJP can be divided into two categories, feature-based approaches (Section 4.1) and neural-based approaches (Section 4.2).

4.1 Feature-based Approaches

In traditional feature-based approaches to LJP, an off-the-shelf learning algorithm such as Support Vector Machines or Random Forests is used to train a classifier (typically a linear classifier) for each LJP subtask [Aletras *et al.*, 2016; Medvedeva *et al.*, 2018; Feng *et al.*, 2018; Katz *et al.*, 2017; Sulea *et al.*, 2017a; Sulea *et al.*, 2017b]. Each training instance corresponds to a legal case in the training set and is typically represented using two types of features. *Lexical features* are composed of word n-grams extracted from the facts of a case [Aletras *et al.*, 2016; Sulea *et al.*, 2017b]. *External features*, on the other hand, encode information that is not included in the facts of a case. For instance, if the case is an appeal, the external features may encode the view of the lower court and the rate of reversal of the court judging the case [Lin *et al.*, 2012; Katz *et al.*, 2017]. Engineering useful external features is a time-consuming and labor-intensive process. Nevertheless, empirical results show that external features can improve LJP performance when used in combination with lexical features.

4.2 Neural-based Approaches

Like many NLP tasks, LJP subtasks have primarily been addressed using neural approaches in recent years.

Embeddings. The n-gram representation used in the non-neural approaches described above is a poor representation of word semantics. To understand why, consider "dog" and "cat". Intuitively, these two words are distributionally similar to each other, but when they are represented as a bag of words, they are essentially one-hot vectors and hence have a similarity of 0. Word embeddings, on the other hand, are a better representation of word semantics than one-hot vectors, as they correspond more closely to what we think of as being similar. Informally, a word embedding is a low-dimensional real-valued vector representation of a word that can be trained so that two words that are semantically similar are close to each other in the embedding space. For instance, "king" and "queen" should have similar embeddings, whereas "king" and "table" should not. While some embeddings can be pre-trained on large corpora via off-the-shelf embedding models (e.g., Word2Vec [Mikolov *et al.*, 2013]), others can be learned during training [Feng *et al.*, 2019].

Like other neural models, neural-based LJP systems encode the words that appear in the text of a case using these word embeddings, which can in turn be exploited by different neural encoders, such as a CNN [Li *et al.*, 2018; Wang *et al.*, 2018], a LSTM [Chen *et al.*, 2019], or a GRU [Paul *et al.*, 2020], to encode the text of a case that better represents its meaning. Since these encoders are not particularly effective at capturing long-distance dependencies, hierarchical encoders have been used to encode cases that are long, where the low-level encoder first encodes snippets of texts (typically sentences) into vectors and the high-level encoder then combines the low-level vectors to create a document representation [Luo *et al.*, 2017]. Regardless of which encoder is used, the encoded input can be fed to a feedforward neural network to predict the label of each instance. As in feature-based approaches, the possible labels for an instance vary depending on the LJP subtask being addressed.

Attention. Some words in a case are more informative than others for a given LJP subtask. However, the aforementioned models do not distinguish between different words by their informativeness. Hence, one way to improve these models would be to give more attention to these informative words. To identify these words in a neural framework, LJP researchers have employed an attention mechanism [Feng *et al.*, 2019; Feng *et al.*, 2021; Li *et al.*, 2019], which attends to all the words in the input sequence and assigns a weight to each word that indicates its importance.

Manually-designed features. To motivate the use of manually-designed features, consider the following example. In Charge prediction, some charge pairs have subtle differences and therefore are *confusing* from the model's point of view. For instance, the definitions of Theft and Robbery only differ in the verification of a specific act (e.g., violence is involved in a Robbery but not a Theft). While an attention mechanism can be used to extract the important keywords specific to each charge, it is far from being able to discriminate such tiny, low-level differences. To address this problem, Hu *et al.* [2018] manually design features that can help distinguish between confusing charges. For instance, to distinguish Theft from Robbery, a useful binary feature would be *Violence*, which encodes whether the crime has an act of violence. After that, they train a model to predict the value of each hand-crafted feature. Finally, the model employs the hidden states of these features in Charge prediction, yielding a 7.7% increase in macro F1 compared to a model that does not employ such features. Manually-designed features have thus far been exploited for Charge prediction only, but we believe they can be similarly explored for other LJP subtasks that have confusing labels, such as Law Article prediction.

Cross-task dependencies. So far, we have only discussed neural approaches in which the LJP subtasks are tackled independently of each other (i.e., a separate model is trained for each LJP subtask). However, there exist logical dependencies among different LJP subtasks that could be profitably exploited by a model. For instance, a law article stipulates a specific prison term if it is violated. In other words, a law article predicted by a LJP system offers the legal basis for Prison Term prediction. The simplest way to exploit this kind of cross-task dependency is to adopt a pipeline approach where law article prediction precedes prison term prediction. However, pipeline approaches are prone to error propagation: if the law article is incorrectly predicted, then basing the prediction of prison term on the predicted law article will likely do more harm than good.

To better exploit cross-task dependencies, several researchers have attempted to design *multi-task* learning frameworks that are applicable to LJP [Zhong *et al.*, 2018; Chen *et al.*, 2020; Yang *et al.*, 2019; Dong and Niu, 2021; Luo *et al.*, 2017; Feng *et al.*, 2019; Huang *et al.*, 2021]. For example, while Zhong *et al.* [2018] predict the different LJP subtasks in a pipeline fashion, they make their model more robust to error propagation by giving as the input to each subtask not only the prediction made for each subtask that appear earlier in the pipeline but also the facts encoded by the encoder. In other words, the prediction for the current subtask is no longer based solely on the (possibly erroneous) predic-

tions for the preceding subtasks. Since Zhong *et al.* [2018] capture cross-task dependencies in the forward direction, a natural question is: can the model be improved by additionally capturing these dependencies in the backward direction? To answer this question, Yang *et al.* [2019] model dependent subtasks bidirectionally, where each subtask's feature representation is first derived from both forward and backward cross-task dependency propagation and then used to predict the result for the subtask. Despite the fact that these models offer improved results, they cannot guarantee that the predictions made for different subtasks are *consistent* with each other. For instance, the predicted charge may not be one of those charges stipulated by the predicted law article. To address this issue, Dong and Niu [2021] propose a graph-based method that employs constraints to ensure that the predictions made for different subtasks are consistent. The use of constraints has been shown to improve LJP subtasks among which cross-task dependencies exist, including Law Article prediction, Charge prediction, and Prison Term prediction.

Within-task label dependencies. The approaches discussed so far have all assumed that the labels in a given LJP subtask are independent of each other. However, for certain LJP subtasks, the labels can be dependent on each other. For instance, the law articles can be grouped into a tree-shaped hierarchy, where some articles have a parent-child relationship. Such label dependencies can potentially be exploited to improve Law Article prediction by casting the task as a two-layer hierarchical classification problem. To model the hierarchical label structure, Wang *et al.* [2019] design a hierarchical network that first predicts the parent law articles and then the corresponding child articles, yielding better prediction results. While label dependencies have thus far been exploited to improve the prediction of law articles, such kind of dependencies exist in other tasks (e.g., Charge prediction) and therefore can be similarly exploited.

Label semantics. The approaches discussed so far have all viewed the labels to be predicted as *atomic*. However, the LJP subtasks are different from many other classification tasks in that the labels have associated text descriptions that contain rich legal knowledge. In Law Article prediction, for instance, each label corresponds to a law article, which is composed of a description that stipulates the circumstances under which the law article is applicable as well as the specific penalties. Exploiting such label semantics could yield better LJP results [Li *et al.*, 2018; Ge *et al.*, 2021]. Below we describe several attempts to exploit label semantics for LJP.

First, label semantics has been applied to discriminate *confusing* law articles (i.e., law articles that are subtly different and therefore are difficult for a system to distinguish). For instance, Xu *et al.* [2020] first construct subgraphs based on the semantics of the law articles such that the law articles that belong to the same subgraph share similar application scenarios. Then they employ a graph network in each subgraph to distill the subtle differences between the law articles, and concatenate the distilled information with the global document vector to make predictions for LJP subtasks.

Second, label semantics has been used to match law articles with facts [Wang *et al.*, 2018; Wang *et al.*, 2019]. Specifically, each law article is matched with each fact in a given case by attending the vector of the law article with the vector of the fact. In essence, the semantics of each law article works as a query to find the most relevant (i.e., best matching) facts. The queried facts are then passed to a dense layer for Law Article prediction. By leveraging the label semantics (i.e., law article definitions), the correlated facts are extracted from a case and the uncorrelated/noisy facts are ignored. Wang *et al.* [2019] show that using only the selected facts for Law Article prediction can offer substantial improvements.

Third, label semantics has been used to separate the facts of a case into different parts. In reality, human judges in Mainland China follow a topological order to decide the verdicts (i.e., law article and charge) and the sentence (i.e., penalty) according to different parts of the facts. Specifically, the judges first decide the charges and the law articles based on the *adjudging* part of the facts. Then, under the premise that the act has constituted a crime, they determine the penalty according to the *sentencing* part of the facts. To simulate this process, Yue *et al.* [2021] employ charge semantics to separate the facts into parts. Using the separated facts for LJP has enabled them to achieve SOTA results on CAIL2018.

Pre-training. Inspired by the success of pre-trained language models (PLMs) on a variety of NLP tasks, researchers have investigated the role of pre-training in LJP. Owing to the differences in writing style and vocabulary between legal and non-legal texts, directly applying off-the-shelf PLMs to LJP subtasks has not yielded satisfactory improvements. As a result, legal-specific PLMs are developed. For English LJP, a BERT-based language model (Legal-BERT) is pre-trained on English legal texts collected from several sources (e.g., legislation, court cases, contracts) [Chalkidis *et al.*, 2020]. For Chinese LJP, a Longformer-based language model, Lawformer, is pre-trained on Chinese civil and criminal legal documents [Xiao *et al.*, 2021]. These legal-specific PLMs can then be fine-tuned for specific LJP tasks.

Interpretable approaches. In court rulings, explanations may be a requirement or even a human right, as the defendants have the rights to know the reasons behind a judge's decision. As far as LJP is concerned, explainability/interpretability is an important issue, as an explanation provided by a LJP system for each of the decisions it makes can increase a user's confidence in the system. LJP researchers have examined interpretability approaches to LJP in which a LJP system is equipped with the ability to explain its decisions. Broadly, these interpretability approaches can be divided into two categories, pre-explanation approaches and post-explanation approaches.

In *pre-explanation* approaches, explanations are first generated and then predictions are made by a system based on these explanations. Chalkidis *et al.* [2021] propose one of the first pre-explanation approaches to predict alleged law articles and the underlying rationales, where the rationales are the paragraphs extracted from the case under consideration. Specifically, the paragraphs that support the judgment are extracted first, based on which the alleged law articles are predicted. A key weakness of this extraction approach to explanation generation is that the logic behind the system's predictions may not be apparent. In particular, laymen may not be able to identify the essential reasons simply from the ex-

Corpus	Language	System	Approach	Evaluation Results (Macro-F1)					
				V	CD	C	LA	ALA	PT
ECHR2019	English	[Chalkidis *et al.*, 2020]	Pretraining	83.2			59.2		
ECHR2021	English	[Chalkidis *et al.*, 2021]	Pretraining					73.7	
CAIL-small	Chinese	[Yue *et al.*, 2021]	Label semantics			90.9	88.8		39.8
CAIL-big	Chinese	[Yue *et al.*, 2021]	Label semantics			80.5	78.1		41.2
ILDC-multi	English	[Malik *et al.*, 2021]	Pretraining		77.79				
ILDC-single	English	[Malik *et al.*, 2021]	Pretraining		76.55				
SJP	Multilingual	[Niklaus *et al.*, 2021]	Pretraining		68.5 (German) 70.2 (French) 59.8 (Italian)				
FCCR	French	[Sulea *et al.*, 2017a]	Feature-based		98.6 (6-class) 95.8 (8-class)				

Table 3: Performance of state-of-the-art LJP systems on the six commonly-used evaluation corpora. We report the results w.r.t subtasks of LJP: Violation (V), Court Decision (CD), Charge (C), Law Article (LA), Alleged Law Article (ALA) and Prison Term (PT). All results are reported in terms of Macro-F1, except the LA subtask in ECHR2019 and the ALA subtask in ECHR2021, where Micro-F1 is employed.

tracted paragraphs. To better uncover the logic behind a LJP system's predictions, Zhong *et al.* [2020a] employ *manually-designed* features that encode the factors that a human judge would typically use when making decisions. For instance, for cases about dangerous driving, the decision made by a human judge will likely be affected by answers to questions such as *Is the case related to traffic?*, *Did an accident occur?* and *Did the party drink alcohol?*. Given this observation, Zhong *et al.* (1) encode each of these yes/no questions as a binary feature whose value is the answer to the corresponding question, (2) train a model to predict the answer to each question, (3) predict charges and law articles based on the answers, and (4) use the answers as explanations for their system's predictions of charges and law articles.

In *post-explanation* approaches, LJP is first executed to obtain a system's prediction p for a given LJP subtask and then explanations are generated given p. Malik *et al.* [2021] propose a post-explanation approach based on a sentence-level schema, where the important sentences in a case that are relevant to a system's prediction p are identified. The importance of a sentence s is identified by *ablation*: if removing s yields a big change in the system's prediction, s is deemed as important (as far as the system's prediction is concerned) and is subsequently included in the explanation. Another line of work involves generating the *court view* (i.e., the human judge's explanation of his/her judgment decision) given p and using it as an explanation for p [Wu *et al.*, 2020b].

Current interpretability approaches to LJP have not been successful. Empirical results show that explanation generation, whether in a pre- or post-explanation manner, would negatively impact LJP results. The reason can be attributed to the fact that LJP and explanation generation are often jointly learned: joint modeling increases learning complexity and allows explanation generation to influence LJP even in post-explanation approaches. Another issue surrounding existing interpretability approaches is that they cannot guarantee that the explanations and the prediction results are consistent.

5 The State of the Art

In this section, we provide an overview of the systems that have achieved SOTA results on the evaluation corpora described in Section 2. These systems are shown in Table 3.

CAIL2018 (Chinese). As can be seen, Prison Term prediction is far from being solved compared to Law Article prediction and Charge prediction. This should perhaps not be surprising given that it is the most challenging LJP subtask (see Table 1). Though not shown in the table, SOTA models do not perform well on low-frequency law articles and charges as all micro-F1 values are significantly larger than their macro-F1 counterparts [Xiao *et al.*, 2021]. Thus, more attention should be given to this unbalanced label problem. Finally, SOTA models achieve better results on CAIL-small than CAIL-big w.r.t. Charge and Law Article prediction. At first glance, this is surprising since the training set of CAIL-big is more than 10 times larger than that of CAIL-small. Nevertheless, the results obtained from these two datasets are not directly comparable since the test sets are not identical.

ECHR2019 and ECHR2021 (English). SOTA results on ECHR2019 are achieved by BERT-based pre-trained models that are pre-trained on a legal corpus from scratch [Chalkidis *et al.*, 2020]. Compared to the BERT model pre-trained on generic texts, even a BERT-small model pre-trained on legal documents yields better performances. These results suggest the effectiveness of pre-training using in-domain documents. For ECHR2021, which is specifically constructed for interpretability in LJP, jointly explaining and predicting judgment results does not yield better results than only predicting judgment results. As noted before, existing interpretability approaches to LJP have not been particularly successful.

SJP (multilingual). The performance of court decision prediction varies across languages: results on Italian are significantly worse than those on German and French. This can be attributed largely to the fact that the Italian dataset is smaller and has a more skewed label distribution. Cross-lingual transfer learning could be exploited to mitigate this problem.

FCCR (French). Results on the 8-class setup are worse than those of the 6-class setup. This is not surprising, as the task difficulty typically increases with the number of labels.

6 Ethical Considerations

As LJP systems provide advice that could have an impact on a human's decision or judgment, there are ethical considera-

tions that should be taken into account when deploying them.

Applying LJP systems. LJP systems should be designed to assist rather than replace legal professionals in their decision-making processes and offer legal consulting suggestions to laymen without much legal knowledge [Tsarapatsanis and Aletras, 2021]. In other words, the actual decisions should still be made by the professionals themselves.

Debiasing data. Legal justice requires that all individuals be treated fairly and equally regardless of their demographics such as nationality, gender, age and region [Leins *et al.*, 2020; Veale and Binns, 2017; Binns, 2018]. Nevertheless, human judges may be biased due to their own belief and ideology, as well as their personal views on racism and sexism. Such biases could be inherited by LJP systems as the systems are trained on datasets in which the decisions/labels were produced by human judges. In fact, empirical results suggest that existing LJP systems exhibit biases w.r.t. gender and region as a result of the underlying datasets in which demographic information is not masked during data construction [Wang *et al.*, 2021; Chalkidis *et al.*, 2019]. While several efforts have begun to debias models (by using an off-the-shelf named entity recognizer to replace all recognized demographic information with insensitive tags and using an adversarial optimizer [Wang *et al.*, 2021; Chalkidis *et al.*, 2019], for instance), there may be other biases that are yet to be discovered. For instance, word embeddings can encode biases, and so can the knowledge acquired by pre-trained models from pre-training datasets. In general, care should be taken to ensure that LJP systems are unbiased, and figuring out how to program and train LJP systems without biases is critical.

Mimicking human reasoning. While the various LJP subtasks concern *accurately* predicting the court's outcomes, from an ethical perspective an equally important issue is whether a system follows the kind of *reasoning* human judges would use in the prediction process. In other words, it is important to not only make the correct predictions but also employ human reasoning in the process. One way to enable a system to mimic the human reasoning process would be to force the system to comply with formal and informal rules that humans need to follow, including both ethical and legal rules. These rules should be considered and embedded in the AI system during development stage [Li and Zhang, 2017; Binns, 2019]. For instance, a LJP system should rely on the definitions of law articles rather than the experience it learned from the annotated data as the basis of its judgment.

A relevant question is: since human judges can be biased, should a LJP system mimic human reasoning, or should it perform logical reasoning instead? We believe that (1) while some judges may be biased, the majority of the judges in a judicial system are not, or else the system would probably have collapsed; and (2) it may not always be possible for a judge to reason in a purely logical manner: the facts and evidences relevant to a case are often incomplete, and subjective interpretation of the available facts and evidences may be needed. Nevertheless, a human judge should ideally perform logical reasoning over incomplete information, avoiding the use of subjective interpretation as much as possible. If this is indeed how the majority of the human judges make decisions, a LJP system that mimics human reasoning will indeed perform logical reasoning with the caveat that subjective interpretation will be used only to fill the gaps created by missing or incomplete information. While in a typical use case, a layman would use a LJP system to predict a human judge's ruling over a given case, we believe that LJP systems can profitably be used by legal professionals as well. For instance, if a lawyer notices that the reasoning involved in a judge's decision on a case deviates from that provided by a LJP system, s/he may investigate whether the judge has employed more subjectivity than is needed in the decision process and may even take into consideration the reasoning provided by the system when framing his/her argument in an appeal.

Building cost-sensitive LJP systems. In LJP, different prediction errors may have different *costs*. For instance, felony charges would lead to capital punishment, whereas misdemeanor charges may simply lead to fines. Therefore mispredicting felony charges could have far more serious consequences than mispredicting misdemeanor charges. We therefore recommend that LJP researchers build cost-sensitive LJP systems that can take this ethical consideration into account.

7 Concluding Remarks

While researchers are making continued progress on LJP despite its difficulty, a natural question is: what are the promising directions for future work?

Event-centric LJP. A case description is typically a long document that contains a lot of information that is irrelevant to judgment (e.g., the criminal's profile). While an attention mechanism can emphasize the relevant words and deemphasize the irrelevant words, it cannot discard the irrelevant information altogether. More specifically, even if each irrelevant word is given a low weight, the presence of a potentially large number of irrelevant words implies that the total amount of weight assigned to irrelevant words will be non-trivial, thus affecting the semantic representation of the case description encoded by the encoder and ultimately system performance. To address this problem, we propose to employ an *event-centric* approach to LJP. Recall that each law article stipulates the kind of events that would violate the article. Hence, the only kind of information relevant to Law Article prediction would be the types of events that are relevant to the law article under consideration. Consequently, we can first identify the events stipulated by the given law article and then make judgment predictions based on the extracted events. To our knowledge, event-centric LJP has not been explored. Nevertheless, there have been recent attempts to create legal datasets annotated with events, which can be exploited to train models for extracting events from legal texts.

Interpretability. Interpretability remains a challenging issue for LJP researchers. While several interpretability approaches have been developed in the context of LJP, one may consider exploiting general interpretability methods, such as LIME [Ribeiro *et al.*, 2016], Layer wise Relevance Propagation [Bach *et al.*, 2015] and Integrated Gradients [Sundararajan *et al.*, 2017], in LJP in a post-explanation manner.

For those who are not interested in low-hanging fruits, we identify two directions for improving interpretability. The

first one involves providing a *visualization* of the events that contribute to the final judgment decision. Specifically, we propose to (1) construct a timeline (i.e., by timestamping each event mentioned in a given case description); (2) construct *narrative chains* [Chambers and Jurafsky, 2009], which make explicit how the events are related to each other around a common protagonist; and (3) associate each event with the violated articles, charges, and/or prison term. This timeline could help users analyze the case and better understand the judgment logic. Key to this approach is the construction of this timeline, which involves many challenging information extraction tasks.

The second direction involves consistency determination. As noted before, existing approaches fail to guarantee that a judgment prediction and the explanation for it are consistent. This renders the generated explanation useless. Determining whether the two are consistent is by no means trivial, but we believe it is a worthwhile direction. One challenge to this consistency determination task is the lack of annotated data. To address this problem, we propose to draw inspirations from the area of argument mining, where researchers have worked on determining how persuasive an argument is. Specifically, we can (1) view the judgment decision and the generated explanation as an "argument", where explanation serves as the premise of the argument and the judgment decision is its conclusion, and then (2) determine how persuasive this argument is using an existing argument persuasiveness system. If this "argument" is persuasive, it should imply that the explanation and the judgment decision are consistent.

Multilinguality. It is not uncommon for LJP researchers to evaluate their approaches on just one dataset/language. For instance, the use of cross-task dependencies has only been evaluated on Chinese datasets, but there is no reason why the same idea cannot be applied to English LJP. To better understand the cross-lingual applicability of an approach, we encourage researchers to evaluate their approaches on different datasets involving multiple languages. Furthermore, given the benefits of learning knowledge of multiple languages in the same framework [Ouyang *et al.*, 2021], it may be promising to train a multilingual model for LJP that can learn from different languages, particularly the different judgment logic used in different languages.

Data and knowledge debiasing. Fairness is essential for LJP systems as legal outcomes could significantly impact individuals. While existing work has verified that the demographic information in the training data can bias a LJP system's decision [Chalkidis *et al.*, 2019; Wang *et al.*, 2021], there could be other biases that have not been discovered. It is therefore important to design debiasing techniques to identify the remaining biases, if any. In addition, many LJP systems employ embeddings and/or PLMs, which could have biases if they were acquired from biased texts. Hence, it is also important to identify and remove such knowledge biases.

Novel pre-training tasks. While researchers have applied legal-specific PLMs that are pre-trained on legal texts to LJP with some successes, the underlying tasks used to pre-train these PLMs are the usual masking-based tasks such as Masked Language Modeling, where the PLMs are asked to restore the masked words. While the PLMs can acquire some lexical knowledge of legal text through these pre-training tasks, they are unlikely to acquire professional legal knowledge. For this reason, we propose to design new pre-training tasks. For instance, we can pre-train PLMs on the legal exam datasets [Zhong *et al.*, 2020c], in which they are asked to answer questions that involve reasoning with legal knowledge, and the resulting PLMs can then be better equipped with logical reasoning skills in the legal domain.

Clustering. While the vast majority of work on LJP has focused on *classification*, we believe that it would be worthwhile to examine clustering approaches that aim to cluster similar cases. Once the cases are clustered (either by supervised or unsupervised methods), we can predict the judgment decisions for a new case based on the judgment decisions of the cases that are most similar to it. A natural question is: what is the advantage of a clustering approach to LJP over a classification approach? To answer this question, consider Law Article prediction in CAIL2018, which involves predicting one of 183 law articles. As the number of labels increases, the difficulty of the corresponding classification task will also increase. In contrast, since a clustering approach focuses on learning the similarity of two cases, it is likely to be less sensitive to the number of labels.

Acknowledgments

We thank the two anonymous reviewers for their helpful comments on an earlier draft of this paper. This work was supported in part by the National Natural Science Foundation of China (No. 61802167) and the US National Science Foundation (Grant IIS-1528037). Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views or official policies, either expressed or implied, of the funding agencies. Chuanyi Li is the corresponding author.

References

[Aletras *et al.*, 2016] Nikolaos Aletras, Dimitrios Tsarapatsanis, Daniel Preotiuc-Pietro, and Vasileios Lampos. Predicting judicial decisions of the European court of human rights: A natural language processing perspective. *PeerJ Computer Science*, 2:e93, 2016.

[Bach *et al.*, 2015] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):e0130140, 2015.

[Binns, 2018] Reuben Binns. Fairness in machine learning: Lessons from political philosophy. In *Proceedings of FAT*, pages 149–159, 2018.

[Binns, 2019] Reuben Binns. Human judgment in algorithmic loops: Individual justice and automated decision-making. *Regulation & Governance*, 2019.

[Chalkidis *et al.*, 2019] Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. Neural legal judgment prediction in English. In *Proceedings of ACL*, pages 4317–4323, 2019.

[Chalkidis *et al.*, 2020] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. LEGAL-BERT: The muppets straight out of law school. In *EMNLP: Findings*, pages 2898–2904, 2020.

[Chalkidis *et al.*, 2021] Ilias Chalkidis, Manos Fergadiotis, Dimitrios Tsarapatsanis, Nikolaos Aletras, Ion Androutsopoulos, and Prodromos Malakasiotis. Paragraph-level rationale extraction through regularization: A case study on European court of human rights cases. In *Proceedings of NAACL-HLT*, pages 226–241, 2021.

[Chambers and Jurafsky, 2009] Nathanael Chambers and Dan Jurafsky. Unsupervised learning of narrative schemas and their participants. In *Proceedings of ACL*, pages 602–610, 2009.

[Chen *et al.*, 2019] Huajie Chen, Deng Cai, Wei Dai, Zehui Dai, and Yadong Ding. Charge-based prison term prediction with deep gating network. In *Proceedings of EMNLP-IJCNLP*, pages 6361–6366, 2019.

[Chen *et al.*, 2020] Wenqing Chen, Jidong Tian, Liqiang Xiao, Hao He, and Yaohui Jin. Exploring logically dependent multi-task learning with causal inference. In *Proceedings of EMNLP*, pages 2213–2225, 2020.

[Dong and Niu, 2021] Qian Dong and Shuzi Niu. Legal judgment prediction via relational learning. In *Proceedings of SIGIR*, pages 983–992, 2021.

[Feng *et al.*, 2018] Yi Feng, Jidong Ge, Chuanyi Li, Li Kong, Feifei Zhang, and Bin Luo. Statutes recommendation using classification and co-occurrence between statutes. In *Proceedings of PRICAI*, pages 326–334, 2018.

[Feng *et al.*, 2019] Yi Feng, Chuanyi Li, Jidong Ge, and Bin Luo. Improving statute prediction via mining correlations between statutes. In *Proceedings of ACML*, pages 710–725, 2019.

[Feng *et al.*, 2021] Yi Feng, Chuanyi Li, Jidong Ge, Bin Luo, and Vincent Ng. Recommending statutes: A portable method based on neural networks. *TKDD*, 15(2):1–22, 2021.

[Gan *et al.*, 2021] Leilei Gan, Kun Kuang, Yi Yang, and Fei Wu. Judgment prediction via injecting legal knowledge into neural networks. In *Proceedings of AAAI*, pages 12866–12874, 2021.

[Ge *et al.*, 2021] Jidong Ge, Yunyun Huang, Xiaoyu Shen, Chuanyi Li, and Wei Hu. Learning fine-grained fact-article correspondence in legal cases. *IEEE/ACM Trans. Audio Speech Lang. Process.*, 29:3694–3706, 2021.

[Hu *et al.*, 2018] Zikun Hu, Xiang Li, Cunchao Tu, Zhiyuan Liu, and Maosong Sun. Few-shot charge prediction with discriminative legal attributes. In *Proceedings of COLING*, pages 487–498, 2018.

[Huang *et al.*, 2021] Yunyun Huang, Xiaoyu Shen, Chuanyi Li, Jidong Ge, and Bin Luo. Dependency learning for legal judgment prediction with a unified text-to-text transformer. *arXiv preprint arXiv:2112.06370*, 2021.

[Katz *et al.*, 2017] Daniel Martin Katz, Michael J. Bommarito, and Josh Blackman. A general approach for predicting the behavior of the supreme court of the United States. *PLOS ONE*, 12(4):e0174698, 2017.

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

[Leins *et al.*, 2020] Kobi Leins, Jey Han Lau, and Timothy Baldwin. Give me convenience and give her death: Who should decide what uses of NLP are appropriate, and on what basis? In *Proceedings of ACL*, pages 2908–2913, 2020.

[Li and Zhang, 2017] Xiuquan Li and Tao Zhang. An exploration on artificial intelligence application: From security, privacy and ethic perspective. In *Proceedings of ICCCBDA*, pages 416–420, 2017.

[Li *et al.*, 2018] Chuanyi Li, Jingjing Ye, Jidong Ge, Li Kong, Haiyang Hu, and Bin Luo. A novel convolutional neural network for statutes recommendation. In *Proceedings of PRICAI*, pages 851–863, 2018.

[Li *et al.*, 2019] Shang Li, Boyang Liu, Lin Ye, Hongli Zhang, and Binxing Fang. Element-aware legal judgment prediction for criminal cases with confusing charges. In *Proceedings of ICTAI*, pages 660–667, 2019.

[Lin *et al.*, 2012] Wan-Chen Lin, Tsung-Ting Kuo, Tung-Jia Chang, Chue-Han Yen, Chao-Ju Chen, and Shou-De Lin. Exploiting machine learning models for Chinese legal documents labeling, case classification, and sentencing prediction. *International Journal of Computational Linguistics & Chinese Language Processing - Special Issue on Selected Papers from ROCLING XXIV*, 17(4), 2012.

[Luo *et al.*, 2017] Bingfeng Luo, Yansong Feng, Jianbo Xu, Xiang Zhang, and Dongyan Zhao. Learning to predict charges for criminal cases with legal basis. In *Proceedings of EMNLP*, pages 2727–2736, 2017.

[Malik *et al.*, 2021] Vijit Malik, Rishabh Sanjay, Shubham Kumar Nigam, Kripabandhu Ghosh, Shouvik Kumar Guha, Arnab Bhattacharya, and Ashutosh Modi. ILDC for CJPE: Indian legal documents corpus for court judgment prediction and explanation. In *Proceedings of ACL*, pages 4046–4062, 2021.

[Medvedeva *et al.*, 2018] Masha Medvedeva, Michel Vols, and Martijn Wieling. Judicial decisions of the European court of human rights: Looking into the crystal ball. In *Proceedings of the Conference on Empirical Legal Studies*, page 24, 2018.

[Mikolov *et al.*, 2013] Tomás Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of NeurIPS*, pages 3111–3119, 2013.

[Nagel, 1963] Stuart S. Nagel. Applying correlation analysis to case prediction. *Tex. L. Rev.*, 42:1006, 1963.

[Niklaus *et al.*, 2021] Joel Niklaus, Ilias Chalkidis, and Matthias Stürmer. Swiss-judgment-prediction: A multilingual legal judgment prediction benchmark. *arXiv preprint arXiv:2110.00806*, 2021.

[Ouyang *et al.*, 2021] Xuan Ouyang, Shuohuan Wang, Chao Pang, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. ERNIE-M: Enhanced multilingual representation by aligning cross-lingual semantics with monolingual corpora. In *Proceedings of EMNLP*, pages 27–38, 2021.

[Paul *et al.*, 2020] Shounak Paul, Pawan Goyal, and Saptarshi Ghosh. Automatic charge identification from facts: A few sentence-level charge annotations is all you need. In *Proceedings of COLING*, pages 1011–1022, 2020.

[Ribeiro *et al.*, 2016] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of ACM SIGKDD*, pages 1135–1144, 2016.

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

[Sulea *et al.*, 2017a] Octavia-Maria Sulea, Marcos Zampieri, Shervin Malmasi, Mihaela Vela, Liviu P. Dinu, and Josef van Genabith. Exploring the use of text classification in the legal domain. In *Proceedings of ASAIL@ICAIL*, volume 2143, 2017.

[Sulea *et al.*, 2017b] Octavia-Maria Sulea, Marcos Zampieri, Mihaela Vela, and Josef van Genabith. Predicting the law area and decisions of French supreme court cases. In *Proceedings of RANLP*, pages 716–722, 2017.

[Sundararajan *et al.*, 2017] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of ICML*, pages 3319–3328, 2017.

[Tsarapatsanis and Aletras, 2021] Dimitrios Tsarapatsanis and Nikolaos Aletras. On the ethical limits of natural language processing on legal text. In *ACL-IJCNLP: Findings*, pages 3590–3599, 2021.

[Urchs *et al.*, 2021] Stefanie Urchs, Jelena Mitrovic, and Michael Granitzer. Design and implementation of German legal decision corpora. In *Proceedings of ICAART (2)*, pages 515–521, 2021.

[Veale and Binns, 2017] Michael Veale and Reuben Binns. Fairer machine learning in the real world: Mitigating discrimination without collecting sensitive data. *Big Data & Society*, 4(2), 2017.

[Wang *et al.*, 2018] Pengfei Wang, Ze Yang, Shuzi Niu, Yongfeng Zhang, Lei Zhang, and ShaoZhang Niu. Modeling dynamic pairwise attention for crime classification over legal articles. In *Proceedings of SIGIR*, pages 485–494, 2018.

[Wang *et al.*, 2019] Pengfei Wang, Yu Fan, Shuzi Niu, Ze Yang, Yongfeng Zhang, and Jiafeng Guo. Hierarchical matching network for crime classification. In *Proceedings of SIGIR*, pages 325–334, 2019.

[Wang *et al.*, 2021] Yuzhong Wang, Chaojun Xiao, Shirong Ma, Haoxi Zhong, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. Equality before the law: Legal judgment consistency analysis for fairness. *arXiv preprint arXiv:2103.13868*, 2021.

[Wu *et al.*, 2020a] Tien-Hsuan Wu, Ben Kao, Anne SY Cheung, Michael MK Cheung, Chen Wang, Yongxi Chen, Guowen Yuan, and Reynold Cheng. Integrating domain knowledge in ai-assisted criminal sentencing of drug trafficking cases. In *Proceedings of JURIX*, volume 334, page 174, 2020.

[Wu *et al.*, 2020b] Yiquan Wu, Kun Kuang, Yating Zhang, Xiaozhong Liu, Changlong Sun, Jun Xiao, Yueting Zhuang, Luo Si, and Fei Wu. De-biased court's view generation with causality. In *Proceedings of EMNLP*, pages 763–780, 2020.

[Xiao *et al.*, 2018] Chaojun Xiao, Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Zhiyuan Liu, Maosong Sun, Yansong Feng, Xianpei Han, Zhen Hu, Heng Wang, and Jianfeng Xu. CAIL2018: A large-scale legal dataset for judgment prediction. *arXiv preprint arXiv:1807.02478*, 2018.

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

[Xu *et al.*, 2020] Nuo Xu, Pinghui Wang, Long Chen, Li Pan, Xiaoyan Wang, and Junzhou Zhao. Distinguish confusing law articles for legal judgment prediction. In *Proceedings of ACL*, pages 3086–3095, 2020.

[Yang *et al.*, 2019] Wenmian Yang, Weijia Jia, Xiaojie Zhou, and Yutao Luo. Legal judgment prediction via multi-perspective bi-feedback network. In *Proceedings of IJCAI*, pages 4085–4091, 2019.

[Yue *et al.*, 2021] Linan Yue, Qi Liu, Binbin Jin, Han Wu, Kai Zhang, Yanqing An, Mingyue Cheng, Biao Yin, and Dayong Wu. NeurJudge: A circumstance-aware neural framework for legal judgment prediction. In *Proceedings of SIGIR*, pages 973–982, 2021.

[Zhong *et al.*, 2018] Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Chaojun Xiao, Zhiyuan Liu, and Maosong Sun. Legal judgment prediction via topological learning. In *Proceedings of EMNLP*, pages 3540–3549, 2018.

[Zhong *et al.*, 2020a] Haoxi Zhong, Yuzhong Wang, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. Iteratively questioning and answering for interpretable legal judgment prediction. In *Proceedings of AAAI*, pages 1250–1257, 2020.

[Zhong *et al.*, 2020b] Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. How does NLP benefit legal system: A summary of legal artificial intelligence. In *Proceedings of ACL*, pages 5218–5230, 2020.

[Zhong *et al.*, 2020c] Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. JEC-QA: A legal-domain question answering dataset. In *Proceedings of AAAI*, pages 9701–9708, 2020.