Semi-Supervised Abductive Learning and Its Application to Theft Judicial Sentencing

Yu-Xuan Huang[1], Wang-Zhou Dai[2], Jian Yang[3], Le-Wen Cai[1],
Shaofen Cheng[4], Ruizhang Huang[3], Yu-Feng Li[1] and Zhi-Hua Zhou[1]
[1]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
[2]Department of Computing, Imperial College London, London SW7 2AZ, UK
[3]College of Computer Science and Technology, Guizhou University, Guiyang 550025, China
[4]Guizhou Higher People's Court, Guiyang, China
Email:[1]{huangyx, cailw, liyf, zhouzh}@lamda.nju.edu.cn [2]w.dai@imperial.ac.uk
[3]Jyang_gzu@163.com [3]rzhuang@gzu.edu.cn [4]shaofencheng@outlook.com

***Abstract*—In many practical tasks, there are usually two kinds of common information: cheap unlabeled data and domain knowledge in the form of symbols. There are some attempts using one single information source, such as semi-supervised learning and abductive learning. However, there is little work to use these two kinds of information sources at the same time, because it is very difficult to combine symbolic logical representation and numerical model optimization effectively. The learning becomes even more challenging when the domain knowledge is insufficient. In this paper, we present an attempt—Semi-Supervised ABductive Learning (SS-ABL) framework. In this framework, semi-supervised learning is trained via pseudo labels of unlabeled data generated by abductive learning, and the background knowledge is refined via the label distribution predicted by semi-supervised learning. The above framework can be optimized iteratively and can be naturally interpretable. The effectiveness of our framework has been fully verified in the theft judicial sentencing of real legal documents. In the case of missing sentencing elements and mixed legal rules, our framework is apparently superior to many existing baseline practices, and provides explanatory assistance to judicial sentencing.**

***Index Terms*—Abductive Learning; Semi-Supervised Learning; Theft Judicial Sentencing**

I. Introduction

Machine learning has achieved great success in a wide variety of tasks. The most commonly used supervised learning paradigm aims at learning a mapping function based on labeled training data. However, labeled data are often hard to obtain due to multiple reasons [1]: 1) Tagging labels for data involves human annotation, which may be very expensive. 2) Popular machine learning algorithms such as deep learning require a huge amount of labeled data, while obtaining them is time-consuming. 3) It usually needs expert knowledge to perform labeling. The concept of Semi-supervised learning (SSL) [1] was introduced to counter these disadvantages, considering that unlabeled data are often cheap and easy to collect. The dataset for SSL contains a small proportion of labeled data and a large amount of unlabeled data. The goal is to use the information from unlabeled data to improve the performance.

Apart from the unlabeled data, in many situations, there exists a great deal of domain knowledge in the form of symbolic rules (e.g., first-order logic formulae). Some efforts have been devoted to utilizing symbolic background knowledge when learning from symbolized data, e.g., Inductive Logic Programming [2], Statistical Relational Learning [3], and Probabilistic Logic Programming [4]. Recently, the latest Abductive Learning (ABL) framework is proposed [5], [6] to bridge supervised learning and symbolic reasoning in the tasks with raw input space and symbolic domain knowledge.

In practice, it is often demanded to leverage both unlabeled data and symbolic background knowledge together. *Theft Judicial Sentencing* (cf. Section V) is a typical task. The task aims to learn an accurate model to extract sentencing elements from criminal judgment documents, and predict the judicial sentence based on those elements. However, the labeling work needs legal professionals and costs much money and time, resulting in a lack of labeled data. On the other hand, there exist a lot of laws as domain knowledge that can only be expressed in the form of symbolic rules.

The major difficulty for exploiting both unlabeled data and symbolic domain knowledge in machine learning lies in the fact that it is difficult for symbolic knowledge to inject into the learning process of common SSL methods. To make use of symbolic rules, the ability of reasoning is needed to conduct logical inference, while SSL algorithms usually involve optimization of numerical values. In addition, popular optimization techniques such as gradient descent cannot directly solve the optimization involving symbolic relations. It is worth mentioning that domain knowledge is usually incomplete, and the learning process also needs to consider this issue.

This paper tackles this challenge by proposing a framework named *Semi-Supervised ABductive Learning* (SS-ABL), in which an SSL model and a symbolic knowledge base can mutually benefit by using logical abduction. It first uses the labeled data to train a model for generating pseudo-labels of the unlabeled data, which may contain many mistakes. Then, SS-ABL explores logical abduction to revise the pseudo-labels based on the symbolic domain knowledge. It tries to abduce pseudo-labels consistent with knowledge base, targeting at correcting the most likely inaccurate pseudo-labels. Then the revised pseudo-labels are used to update the SSL model. The same circulation repeats until it reaches the turn limit.

2374-8486/20/$31.00 ©2020 IEEE
DOI 10.1109/ICDM50108.2020.00127

We verify the SS-ABL framework in a real-world task of *Theft Judicial Sentencing*. The dataset consists of criminal judgment texts of theft, and only a small proportion are labeled. At the same time, the knowledge base contains a large number of symbolic rules extracted from laws. Experimental results show that SS-ABL can make full use of the domain knowledge and unlabeled data, leading to an improvement of classifier as well as interpretable sentence prediction rules, even when there is only a handful of labeled data and insufficient domain knowledge.

II. Related Work

A wide range of powerful approaches have emerged to deal with SSL, such as generative modeling [7], graph-based methods [8], disagreement-based methods [9], [10], low-density methods [11]–[13], regularization-based methods [14]–[16]. Recently, there are also some studies considering the SSL methods in noisy environments [17], [18]. In the following, we focus on low-density and regularization-based methods that are mostly related to our work. For more comprehensive overviews on SSL, refer to [1].

Low-density methods [1], [11]–[13] are proposed based on a basic assumption that "the decision boundary should lie in low-density regions" [1]. TSVM [11] is a classical SSL method for SVM, which seeks a separating hyperplane maximizing margin for all samples. Entropy Minimization [12] adds a loss term that encourages the classifier to make high confident predictions for unlabeled examples. In this case, "confident" prediction is equivalent to low entropy, meaning that samples should be away from the decision boundary. Pseudo-Label [13], a kind of self-training [19] algorithm that imputes the labels of high confident examples, has shown it fantastic ability. In fact, as mentioned in [13], it can be regarded as an implicit entropy regularization, which encourages the predicted probabilities of the unlabeled data to be near 1-of-K code, so that the entropy is minimized.

Regularization-based methods [14]–[16] are successful methods mainly based on the assumption that "examples with perturbations share the same labeling". In the neural network community, a number of SSL methods based on consistency regularization try to make perturbation in various ways. Π-Model and Temporal Ensembling [14] maintain an exponential moving average of label predictions, and encourages the distance between this target and current output to be small. Mean Teacher [15] tries to obtain a more stable output by averaging model weights instead of label predictions. Virtual Adversarial Training (VAT) [16] tries to directly find the adversarial direction which can most significantly affect the output. From the Bayesian standpoint, a regularization term can be interpreted as a prior distribution that reflects a priori knowledge [20]. The methods such as Π-Model, Mean Teacher, VAT hold the apriori belief that the outputs should be smooth in regard to spatial or temporal inputs.

Some approaches to utilize symbolic rules in machine learning have been put forward. Methods involving neural networks [21], [22] try to focus on relations or differentiable knowledge representations, but their behaviors are hard to interpret. Probabilistic Logic Program (PLP) [4] and Statistical Relational Learning (SRL) [3] aim at integrating machine learning and logical reasoning by preserving the symbolic representation. However, they usually require direct semantic-level input. Abductive learning [5], [6] is able to preserve the interpretability of symbolic rules while processing sub-symbolic data at the same time.

Our proposed SS-ABL framework can be regarded as a combination of Pseudo-Label and implicit regularization methods. Pseudo-Label enforces unlabeled samples to be away from the decision boundary. The revision of pseudo-labels based on abductive reasoning serves as an implicit regularization term to utilize domain knowledge. The revision process incorporates a priori knowledge expressed as rules and tries to make pseudo-labels consistent with knowledge base. If unlabeled samples fall in low-density regions and get incorrect pseudo-labels, then a priori knowledge expressed as rules helps to decide the correct labels. Therefore, its purpose is exactly the same as the aim of other common regularization methods.

III. Abductive Learning

ABductive Learning (ABL) [5], [6] is a novel framework that unifies two AI paradigms—machine learning and logical reasoning—in a mutually beneficial way. In ABL, the machine learning model learns to perceive primitive logic facts from raw data, while logical abduction exploits symbolic domain knowledge and corrects the wrongly perceived facts for improving the machine learning models.

Abduction (i.e., *abductive reasoning*) is one of the three basic forms of logic inference, while the others are *deduction* and *induction* [23]. *Deduction* refers to reasoning from general rules to special cases, and *induction* denotes inferring general rules from special cases. Different from *deduction* and *induction*, *abduction* means forming a ground hypothesis that explains observed phenomena.

To illustrate the idea more clearly, this paper denotes logical symbols as follows: "$\neg$" is negation (not); "$\wedge$" is conjunction (and); "$\vee$" is disjunction (or); "$\leftarrow$" is implication, which means that if the condition on the right of "$\leftarrow$" holds, then the consequent on the left holds. For example, consider the following logical rules:

$$wet\_grass \leftarrow rain\_last\_night \vee sprinkler\_was\_on, \quad (1)$$

$$wet\_shoes \leftarrow wet\_grass, \quad (2)$$

$$false \leftarrow rain\_last\_night \wedge sprinkler\_was\_on, \quad (3)$$

where the first two formulas state the causes for grass and shoes being wet, and the last formula specifies that $rain\_last\_night$ and $sprinkler\_was\_on$ cannot be true at the same time. When we observe wet_shoes, rule (2) indicates that wet_grass should also be true. Continuing this process, based on rule (1), both $rain\_last\_night$ and $sprinkler\_was\_on$ are two possible explanations. If we also observe that no rain occurred last night, according to (3), $sprinkler\_was\_on$ would be the only explanation.

IV. The Semi-Supervised Abductive Learning Framework

In this section, we introduce the SS-ABL framework. We first present the problem setting and then provide an overview of the framework, followed by optimization details.

A. Problem Setting

In supervised learning, we are given labeled data $X = \{x_1, x_2, ..., x_l\}$, and their corresponding ground-truth labels $Y = \{y_1, y_2, ..., y_l\}$. The task is to learn a function $f : X \mapsto Y$, which would give correct output over unseen data. In SSL, we are provided additionally with unlabeled data $X_u = \{x_{l+1}, x_{l+2}, ..., x_{l+u}\}$, usually $l << u$. The unlabeled data are utilized to improve performance of function f.

When it comes to the setting of SS-ABL, the input data still contain labeled data X_l and their ground-truth label Y_l. But additionally, we have unlabeled data X_u and a knowledge base KB_θ. The incomplete KB_θ consists of a number of first-order symbolic rules, as well as unknown parameters θ (e.g., weights of the rules). The goal of learning is to utilize all labeled and unlabeled data to train a classifier with the help of KB_θ and optimize parameters θ. Fig. 1 compares settings of supervised learning, SSL and SS-ABL.

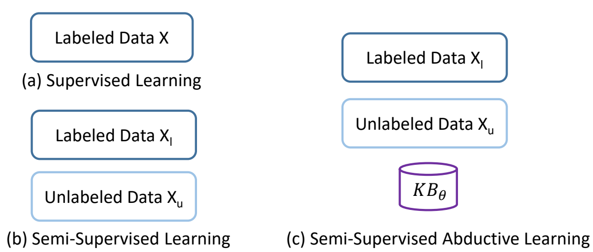

Fig. 1. The setting of SS-ABL.

B. Framework

We propose the *Semi-Supervised ABductive Learning* (SS-ABL) framework, which combines logical abduction with SSL to exploit both domain knowledge and unlabeled data.

SS-ABL involves the Pseudo-Labeling technique [13], which uses unlabeled data to improve performance by regarding their high-confident pseudo-labels as ground-truth labels. However, when labeled data are limited or low-quality, the model may produce incorrect pseudo-labels, because it is difficult for the model to learn the latent cluster structure in this situation [24]–[26]. Therefore, pseudo-labels may be unreliable in many tasks that cannot obtain enough high-quality labels.

Intuitively, SS-ABL first obtains pseudo-labels from a classifier trained by labeled data. The pseudo-labels may be incorrect and inconsistent with knowledge base. SS-ABL tries to revise the pseudo-labels by abductive reasoning. The abductive reasoning module abduces the most likely correct label according to knowledge base. Then the revised pseudo-labels will be used as ground-truth labels in the following training process. At the same time, parameters in knowledge base are updated based on pseudo-labels. After updating the classifier and parameters, the above routine repeats iteratively. Finally, we obtain a classifier and a knowledge base. Fig. 2 illustrates the framework of SS-ABL.

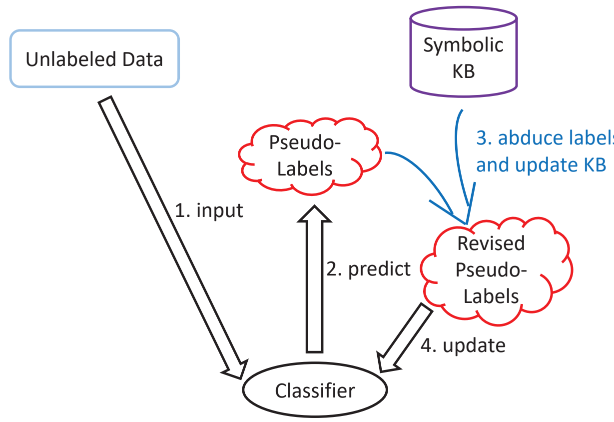

Fig. 2. The framework of SS-ABL.

Formally, we define SS-ABL as follows. Given labeled data X_l, Y_l, unlabeled data X_u and knowledge base KB_θ, seek function f and parameters θ that minimize the objective:

$$\begin{aligned} &\frac{1}{N_l}\sum_{i=1}^{N_l} L(y_i, f_i) + \alpha(t)\frac{1}{N_u}\sum_{i=1}^{N_u} L(\Delta(y_i'), f_i') \\ &- \beta\frac{1}{N_u}\sum_{i=1}^{N_u} Con(\Delta(y_i'), KB_\theta), \end{aligned} \tag{4}$$

where N_l is the number of labeled data, N_u for unlabeled data, f_i is the function's output of labeled data, y_i is the ground-truth label, f_i' for unlabeled data, y_i' is the pseudo-label, $\Delta(y_i')$ is the revised pseudo-label by logical abduction process Δ, which will be explained later. L is the loss function, $\alpha(t)$ and β are balancing coefficients. Con stands for a function that outputs how consistent is $\Delta(y_i')$ with KB_θ.

The first term $\frac{1}{N_l}\sum_{i=1}^{N_l} L(y_i, f_i)$ is the loss on the labeled data. The second term $\alpha(t)\frac{1}{N_u}\sum_{i=1}^{N_u} L(\Delta(y_i'), f_i')$ is the loss on revised pseudo-labels of unlabeled data. The first two terms are similar to the semi-supervised loss function in Pseudo-Label [13]. The last term $\beta\frac{1}{N_u}\sum_{i=1}^{N_u} Con(\Delta(y_i'), KB_\theta)$ indicates the consistency between revised pseudo-labels and knowledge base (the more consistent, the higher its value). Besides revising the inconsistent pseudo-labels, this objective also helps knowledge base learn accurate parameters θ.

The coefficient β is a constant and makes no difference since SS-ABL optimizes the last term separately (see next subsection). $\alpha(t)$ slowly increases as training goes on. It is set based on the schedule in [13].

$$\alpha(t) = \begin{cases} 0 & t < T_1 \\ \frac{t-T_1}{T_2-T_1}\alpha_f & T_1 \le t < T_2 \\ \alpha_f & T_2 \le t \end{cases} \tag{5}$$

Algorithm 1 Semi-Supervised Abductive Learning

Input: Labeled data and their labels X_l, Y_l; Unlabeled data X_u; Knowledge base KB_θ

Output: Function f; Parameters θ

$f \leftarrow$ TrainModel(X_l, Y_l)

$\theta \leftarrow$ TrainParameter(Y_l) # Pretrain model and parameter

while $t < turn\ limit$ **do**

  $Y_u \leftarrow f(X_u)$ # Generate pseudo-labels Y_u

  $\Delta(Y_u) \leftarrow$ Abduce(KB_θ, Y_u) # Revise pseudo-labels

  $\theta \leftarrow$ TrainParameter$(Y_l, \Delta(Y_u))$ # Update parameters

  $f \leftarrow \psi(f, X_u, \Delta(Y_u), X_l, Y_l)$ # Update function f

  $t \leftarrow t + 1$

end while

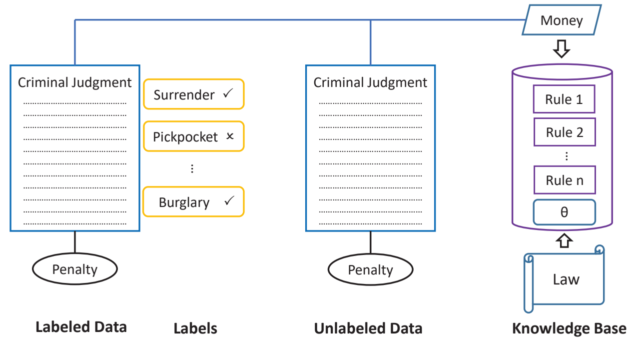

Fig. 3. Dataset and knowledge base of the task.

C. Optimization

The objective (4) involves optimization on symbolic relations, rather than pure numerical optimization which can be solved by methods such as gradient descent. The optimization goal contains two components: function f and knowledge base's parameters θ, which are mutually dependent: parameters θ are calculated based on the output of function f; abductive reasoning requires knowledge base KB_θ where accurate parameters will lead to high-quality revised pseudo-labels. Therefore, SS-ABL tries to optimize them alternatively. Parameters θ in KB_θ are first updated based on pseudo-labels, corresponding to optimizing the last term in the loss function (4). Then the function f is optimized using revised pseudo-labels generated by the abduction of KB_θ, involving the first two terms in (4). An outline of the proposed optimization algorithm is shown in Algorithm 1.

As for how to abduce labels, SS-ABL needs to obtain $\Delta(y)$ most consistent with knowledge base KB_θ. The logical abduction procedure is conducted by assuming that some positions of pseudo-labels are incorrect. In other words, some pseudo-labels are fixed, and the others are made to be "unknown" (abducible). Then the abduction module will abduce most compatible labels and use them to replace pseudo-labels. According to the definition of abductive learning [5], [6], the fixed pseudo-labels serve as observations, and knowledge base KB_θ tries to abduce consistent explanations for other "unknown" pseudo-labels. When the search space is large, we can solve it with derivative-free optimization [5], [27], otherwise we can directly try all hypotheses.

V. Experiments

In this section, we carried out the experiments of *Theft Judicial Sentencing* task to demonstrate that SS-ABL is able to leverage unlabeled data and symbolic knowledge. The code is available for download[1].

A. Task

The task aims to train a classifier that outputs sentencing elements of criminal judgment, as well as optimize the parameters in knowledge base to predict the sentence of a defendant. A criminal judgment is the judicial documents written by the judge in court. The dataset contains 687 court records of theft happened in Guizhou, China in 2017–2018. We focus on theft because they account for a sizeable proportion of all cases. In real life, courts are often required to check the quality of sentences made by magistrates (judges in the local court).

A typical theft case consists of three major parts: 1) the amount of money involved; 2) sentencing elements; 3) the final sentence. The amount of money and the final sentence are numerical values written directly in criminal judgments. Sentencing elements decide whether to give the criminal a heavier or lesser punishment, including recidivism, burglary, etc. They can be found in the text of criminal judgments, but written in various ways rather than a fixed form. Because of the variety of ways to express those elements, a machine learning model is needed to extract them automatically.

Apart from the dataset, there is a lot of domain knowledge that we can exploit in law, which can be written to first-order symbolic rules. Note that part of the rules may be incomplete. For example, we know a law rule that surrender leads to a lesser punishment, but we do not know how much the sentence will be lessened, i.e., parameters θ of rules are unknown.

Figure 3 illustrates the dataset and knowledge base of the task. The dataset contains labeled data and unlabeled data, and labeled data have labels of sentencing elements for each criminal judgment. Each sample, whether labeled or not, has its corresponding amount of money involved and the final sentence, since it is easy to get them directly from criminal judgment texts. The value of money will be used by knowledge base KB_θ, which can be regarded as part of it. The knowledge base contains a large number of symbolic rules, as well as θ which are the learnable numerical parameters of some rules.

The difficulties of the task lie in insufficient high-quality labels and the uncertainty in laws. It uses obfuscated data, which are in fact inadequate. Moreover, the labeling work requires teaching data labelers the professional knowledge of law. As a result, we find that there are still some errors among the labels. Thus labeled data are limited and even inaccurate. Furthermore, though there are a large number of symbolic law rules, they are complex and even some rule parameters are unknown. Therefore, the task is quite challenging for conventional machine learning approaches.

[1] https://github.com/AbductiveLearning/SS-ABL

Domain knowledge for calculating the punishment in sentencing:

$$penalty(X,Y) \leftarrow base\_penalty(X,Z_1) \wedge weight(X,Z_2) \wedge Y = Z_1(1+Z_2).$$
$$base\_penalty(X,Y) \leftarrow money(X,m) \wedge Y = 0.7m + 5.7.$$
$$weight([],0) \leftarrow$$
$$weight([X|Xs],Y) \leftarrow element\_weight(X,Z_1) \wedge weight(Xs,Z_2) \wedge Y = Z_1 + Z_2.$$

Punishment weights for different sentencing elements:

$$element\_weight(recidivism, 29\%). \qquad element\_weight(confession, -11\%).$$
$$element\_weight(pickpocket, 14\%). \qquad element\_weight(burglary, 3\%).$$

Fig. 4. Sample Sentence rules in the knowledge base.

B. Knowledge Base

The knowledge base KB_θ mainly consists of first-order logical rules, including law rules, matching rules and common sense constraints. The amount of money involved in each case and the final sentence of imprisonment are also included in the form of logical facts. Law rules are extracted from *Criminal Law* and related legal documents. Matching rules are about relations between elements and words in text. Common sense constraints encode the relations between sentencing elements.

Figure 4 gives an example of sentencing law rules in knowledge base KB_θ. The four rules on the top are the law rules about how money and elements influence the final penalty. The first formula indicates that it first calculates the base penalty, and gives a lesser or heavier punishment based on weights of sentencing elements. The literal "$penalty(X,Y)$" means document X corresponds to sentence Y, and similar meaning for "$base\_penalty(X,Z_1)$" and "$weight(X,Z_2)$". The "$Y = Z_1(1+Z_2)$" defines how to calculate the final sentence. The four rules on the bottom are punishment weights for different criminal elements. The parameters of this knowledge base include "0.7", "5.7", "29%", "-11%", "14%", "3%'.

C. Experimental Setup

We employ the multi-label Bidirectional Encoder Representations from Transformers (BERT)[2] [28] as the classifier, a state-of-the-art model for Natural Language Processing (NLP). We compare SS-ABL with three types of baselines: 1) BERT that use only labeled data; 2) Abductie learning (ABL) [5], [6] that exploit symbolic rules; 3) SSL methods that leverage unlabeled data, including Pseudo-Label (PL) [13], Tri-training (Tri) [10]. There are three training settings: 10%, 50% and 100% labeled data, denoted as suffixes such as "SS-ABL-10".

The dataset consists of 90% training and 10% test data. In the experiments, all methods share the same structured BERT model and the same rules. We use the default hyperparameters of BERT model in the source code. The hyperparameter M (maximum number of revised label) of SS-ABL is set to 2, and $\alpha_f = 3, T_1 = 100, T_2 = 800$, which is determined by cross-validation on training data. Experiments are repeated five times on a workstation with an Intel Xeon CPU @ 2.60GHz, 32 GB memory and an Nvidia GeForce RTX 2080 Ti GPU. For more details on dataset and experiment, please refer to our code.

[2]We use the official implementation and pre-trained model at https://github.com/google-research/bert

TABLE I
F1-SCORE OF BERT, AND MAE & MSE OF PREDICTED SENTENCE.

Method	F1	MAE	MSE
BERT-10	0.811±0.010	0.867±0.032	1.204±0.123
PL-10	0.814±0.006	0.862±0.035	1.155±0.107
Tri-10	0.812±0.016	0.840±0.066	1.155±0.107
ABL-10	0.824±0.014	0.873±0.102	1.376±0.217
SS-ABL-10	**0.862±0.005**	**0.824±0.017**	**1.146±0.049**
BERT-50	0.857±0.006	0.830±0.020	1.065±0.044
PL-50	0.858±0.010	0.832±0.035	1.045±0.154
Tri-50	0.861±0.007	0.810±0.021	0.994±0.046
ABL-50	0.860±0.003	0.842±0.029	1.082±0.110
SS-ABL-50	**0.865±0.007**	**0.788±0.027**	**0.959±0.091**
BERT-100	0.863±0.003	0.821±0.014	1.011±0.031
ABL-100	0.867±0.008	0.822±0.030	1.007±0.093

D. Results

a) Sentencing Elements Prediction: The first target is to predict the sentencing elements in criminal judgments. Table I shows the F1-score (Micro-averaging) of the multi-label BERT on the test set. The performance of SS-ABL-10 and SS-ABL-50 is superior to other methods. When the label rate is low, the gain from utilizing both unlabeled data and symbolic domain knowledge is much higher. Note that the performance of SS-ABL-10 is even superior to BERT-50 method that uses more labeled data but no unlabeled data. Likewise, the F1-score of SS-ABL-50 has nearly the same level as BERT-100.

b) Judicial Sentence Prediction: The second target involves predicting the sentence of theft cases using the learned model and knowledge base. Table I shows the Mean Absolute Error (MAE) and Mean Square Error (MSE) between the actual and predicted value. According to the results, the SS-ABL-based approaches significantly outperform other compared methods. SS-ABL-based methods make more significant improvements because they can jointly optimize the parameters in the domain knowledge base and the BERT model.

TABLE II
SENTENCE RULES LEARNED BY SS-ABL.

Base sentence=0.64m+6 (m: money involved in thousands)	
burglary $\rightarrow$ base +6%	*confession* $\rightarrow$ base -14%
forgiveness $\rightarrow$ base -15%	*juvenile* $\rightarrow$ base -15%
no loss $\rightarrow$ base -7%	*pickpocket* $\rightarrow$ base +15%
surrender $\rightarrow$ base -10%	*recidivism* $\rightarrow$ base +27%

Although BERT-100 has a similar F1-score as SS-ABL-50, its MAE and MSE are worse. This is probably due to the noise in labeled data. SS-ABL can correct the label noise that mostly influences sentence prediction by considering the legal knowledge, so that the sentence prediction is more accurate.

The learned sentence rules and their parameters are listed in Table II. The rules explain how knowledge base reasons to produce the final sentence prediction. These highly interpretable rules can be easily understood and examined by human, providing explanatory assistance to judge sentencing.

VI. CONCLUSION

In this paper, we propose *Semi-Supervised Abductive Learning*, a framework aiming at combining the symbolic knowledge and semi-supervised learning. In this framework, an SSL model is used for generating pseudo-labels on the unlabeled data, and logical abduction revises those noisy pseudo-labels according to domain knowledge. There are many tasks involving both logical rules and unlabeled data. Theft Judicial Sentencing is a typical challenging task, which needs to tackle the scarcity of labels and the uncertainty in laws. As shown by the experimental results, SS-ABL demonstrates its excellent capability in exploiting the symbolic knowledge and unlabeled data. It achieves significant performance improvement compared with previous approaches that only leverage single information. Besides, SS-ABL is a general-purpose framework with sufficient flexibility. The BERT classifiers can be replaced by other classifiers such as deep forest [29], and rules in the knowledge base are not limited to law rules.

ACKNOWLEDGMENT

This research was supported by the National Key R&D Program of China (2018YFB1004300) and NSFC (61751306, 61921006). The corresponding authors for this work are Yu-Feng Li, Ruizhang Huang and Zhi-Hua Zhou.

REFERENCES

[1] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning*. The MIT Press, 2006.

[2] S. Muggleton, "Inductive logic programming," *New generation computing*, vol. 8, no. 4, pp. 295–318, 1991.

[3] D. Koller, N. Friedman, S. Džeroski, C. Sutton, A. McCallum, A. Pfeffer, P. Abbeel, M.-F. Wong, D. Heckerman, C. Meek *et al.*, *Introduction to statistical relational learning*. The MIT Press, 2007.

[4] L. De Raedt and A. Kimmig, "Probabilistic (logic) programming concepts," *Machine Learning*, vol. 100, no. 1, pp. 5–47, 2015.

[5] W.-Z. Dai, Q. Xu, Y. Yu, and Z.-H. Zhou, "Bridging machine learning and logical reasoning by abductive learning," in *Advances in Neural Information Processing Systems 32 (NeurIPS)*, 2019, pp. 2811–2822.

[6] Z.-H. Zhou, "Abductive learning: towards bridging machine learning and logical reasoning," *Science China Information Sciences*, vol. 62, no. 7, p. 76101, 2019.

[7] B. M. Shahshahani and D. A. Landgrebe, "The effect of unlabeled samples in reducing the small sample size problem and mitigating the hughes phenomenon," *IEEE Transactions on Geoscience and Remote sensing*, vol. 32, no. 5, pp. 1087–1095, 1994.

[8] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proceedings of the 20th International Conference on Machine learning (ICML)*, 2003, pp. 912–919.

[9] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT)*, 1998, pp. 92–100.

[10] Z.-H. Zhou and M. Li, "Tri-training: Exploiting unlabeled data using three classifiers," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 11, pp. 1529–1541, 2005.

[11] T. Joachims, "Transductive inference for text classification using support vector machines," in *Proceedings of the 16th International Conference on Machine Learning (ICML)*, 1999, pp. 200–209.

[12] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *Advances in Neural Information Processing Systems 18 (NIPS)*, 2005, pp. 529–536.

[13] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *ICML Workshop on Challenges in Representation Learning*, 2013.

[14] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," in *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017.

[15] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in Neural Information Processing Systems 30 (NIPS)*, 2017, pp. 1195–1204.

[16] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 8, pp. 1979–1993, 2018.

[17] Z.-Y. Zhang, P. Zhao, Y. Jiang, and Z.-H. Zhou, "Learning from incomplete and inaccurate supervision," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, 2019, pp. 1017–1025.

[18] L.-Z. Guo, Z.-Y. Zhang, Y. Jiang, Y.-F. Li, and Z.-H. Zhou, "Safe deep semi-supervised learning for unseen-class unlabeled data," in *Proceedings of the 37th International Conference on Machine learning (ICML)*, 2020.

[19] H. Scudder, "Probability of error of some adaptive pattern-recognition machines," *IEEE Transactions on Information Theory*, vol. 11, no. 3, pp. 363–371, 1965.

[20] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[21] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap, "A simple neural network module for relational reasoning," in *Advances in Neural Information Processing Systems 30 (NIPS)*, 2017, pp. 4967–4976.

[22] A. Graves, G. Wayne, M. Reynolds *et al.*, "Hybrid computing using a neural network with dynamic external memory," *Nature*, vol. 538, no. 7626, pp. 471–476, 2016.

[23] J. Buchler, *Philosophical writings of Peirce*. Dover, 1955.

[24] Y.-F. Li and D.-M. Liang, "Safe semi-supervised learning: a brief introduction," *Frontiers of Computer Science*, vol. 13, no. 4, pp. 669–676, 2019.

[25] Y.-F. Li, L.-Z. Guo, and Z.-H. Zhou, "Towards safe weakly supervised learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[26] A. Oliver, A. Odena, C. A. Raffel, E. D. Cubuk, and I. Goodfellow, "Realistic evaluation of deep semi-supervised learning algorithms," in *Advances in Neural Information Processing Systems 31 (NeurIPS)*, 2018, pp. 3235–3246.

[27] Y. Yu, H. Qian, and Y.-Q. Hu, "Derivative-free optimization via classification," in *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI)*, 2016, pp. 2286–2292.

[28] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[29] Z.-H. Zhou and J. Feng, "Deep forest," *National Science Review*, vol. 6, no. 1, pp. 74–86, 2019.