Computers, Materials & Continua CMC, vol.62, no.3, pp.1217-1231, 2020

Prison Term Prediction on Criminal Case Description with Deep Learning

Shang Li[1], Hongli Zhang[1,*], Lin Ye[1], Shen Su[2], Xiaoding Guo[1], Haining Yu[1,3] and Binxing Fang[1]

Abstract: The task of prison term prediction is to predict the term of penalty based on textual fact description for a certain type of criminal case. Recent advances in deep learning frameworks inspire us to propose a two-step method to address this problem. To obtain a better understanding and more specific representation of the legal texts, we summarize a judgment model according to relevant law articles and then apply it in the extraction of case feature from judgment documents. By formalizing prison term prediction as a regression problem, we adopt the linear regression model and the neural network model to train the prison term predictor. In experiments, we construct a real-world dataset of theft case judgment documents. Experimental results demonstrate that our method can effectively extract judgment-specific case features from textual fact descriptions. The best performance of the proposed predictor is obtained with a mean absolute error of 3.2087 months, and the accuracy of 72.54% and 90.01% at the error upper bounds of three and six months, respectively.

Keywords: Neural networks, prison term prediction, criminal case, text comprehension.

1 Introduction

For the past few years, the amount of data in the judicial field has grown rapidly. The data involves various legal cases, supplementary extensions of the law and judicial interpretations. Legal professionals, such as judges, lawyers and prosecutors, not only have to handle numerous cases, but also need to consult a large number of files for reference or analyze the data related to the case. It leads to a growing burden on law professionals, which may result in a lower efficiency and an increased risk of making mistakes in the judicial work. To help safeguard judicial fairness and public security, a legal assistant system based on information technology (e.g., artificial intelligence and data mining) should be employed to facilitate the judgment of legal cases.

The task of prison term prediction (PTP) differs from the charge prediction task that, instead of aiming to determine appropriate charges (e.g., the crime of theft, fraud, robbery and intentional injury) for a given case, its objective is to predict the term of penalty (e.g.,

[1] School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China.

[2] Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou, China.

[3] Department of Computer Science, City University of Hong Kong, Kowloon Tong, Hong Kong.

[*] Corresponding Author: Hongli Zhang, Email: zhanghongli@hit.edu.cn.

fixed-term imprisonment counted by year/month, life imprisonment or death penalty) for a certain type of criminal case by analyzing the textual fact description. In mainland China which is one of civil law jurisdictions, courts deal with legal cases based on statutory laws and the fact description, rather than with reference to decisions of precedent cases. The judge will make a final decision by combining the analysis of specific situation of current case with the understanding and interpretation of relevant law articles. Although we can expect a traditional classification model by learning previous similar cases to play a role in the legal assistant system, it is always more convincing to make the prediction with legal basis. However, it is not trivial to train a machine judge to predict appropriate prison term based on law articles and fact descriptions. There are two crucial issues to be addressed: 1) how to effectively extract features well representing a case from textual fact descriptions, and 2) how to implement a refined model which outputs an integral number as the prediction result of prison term.

The majority of existing works attempt to resolve the judgment prediction task by formalizing it as a text classification problem. These efforts either employ off-the-shelf classification models [Hachey and Grover (2006); Goncalves and Quaresma (2005); Palau and Moens (2018)] with shallow features extracted from text [Liu, Chang and Ho (2004); Liu and Hsieh (2006)] or case profiles [Katz, Bommarito II and Blackman (2017)], or attain deeper semantic understanding of case descriptions by manually annotating cases and designing specific features [Lin, Kuo and Chang (2012)]. Despite the introduction of machine learning and natural language processing (NLP) methods that can advance the analysis of legal texts [Xiong, Shen, Wang et al. (2018)], while it remains unsolved to learn better semantic representations from case fact descriptions with less human annotations and make refined prediction of the prison term for a case with a certain charge.

In this paper, we aim to address the PTP problem by incorporating appropriate mechanisms to integrate the textual fact descriptions of criminal cases with legal basis. To obtain a better understanding and more specific representation of the legal texts, we first summarize the corresponding judgment model through comprehensive analysis of relevant law articles and the structure of judgment document. Then we employ state-of-the-art neural network models to build sentence-level multiple binary classifiers, each of which focusing on a specific feature based on the judgment model. After merging sentence-level features into a case-level feature, we adopt the linear regression model and the neural network model to solve the PTP problem. For experiments, we collect and construct a real-world dataset containing more than 40,000 judgment documents of theft cases published by the Supreme People's Court of the People's Republic of China. Experimental results demonstrate that our method can effectively extract judgment-specific case features from textual fact descriptions. The proposed predictor obtains the best performance of 3.2087 months in mean absolute error, and 72.54% and 90.01% in accuracy when the error upper bound being set to three and six months, respectively.

The contributions of this paper are summarized as follows:

1) A two-step deep learning method is proposed to address the PTP problem by integrating the textual fact descriptions of criminal cases with legal basis;
2) To obtain a better understanding and more specific representation of the legal texts,

the judgment model is summarized and then applied in the extraction of case feature from judgment documents;

3) We build a real-world dataset of theft case judgment documents. Experimental results on this dataset demonstrate the effectiveness of our method.

The rest of this paper is organized as follows. Section 2 briefly reviews the related work. The judgment model of theft cases is described in Section 3. In Section 4, we propose the methods of case feature extraction and prison term prediction. Experimental results are presented in Section 5. Finally, Section 6 contains the concluding remarks.

2 Related work

The research of judgment prediction has attracted increasing attention in recent years. Relevant issues in the field of artificial intelligence and law have been studied as well.

In earlier studies on judgment prediction, most researchers tended to formalize it as a text classification problem. Hachey et al. [Hachey and Grover (2006)] proposed a method of classifying legal sentences for automatic court rulings. The work of Goncalves et al. [Goncalves and Quaresma (2005)] was to classify legal text in 3,000 categories based on a taxonomy of legal concepts, and reported a F1 score of 79%. Liu et al. [Liu, Chang and Ho (2004)] presented a case-based reasoning system and used KNN model to classify 12 common criminal charges in Taiwan. The work by Katz et al. [Katz, Bommarito II and Blackman (2017)] built randomized trees with features extracted from case profiles and reported an accuracy of 70.9% in predicting the US Supreme Court's behavior. The work in Lin et al. [Lin, Kuo and Chang (2012)] exploited machine learning methods to identify robbery and intimidation cases and predict their sentencing by considering manually defined 21 legal factor labels. More recently, the work of Aletras et al. [Aletras, Tsarapatsanis, Preoţiuc-Pietro et al. (2016)] aimed to predict decisions of the European Court of Human Rights (ECHR), and they reported an accuracy of 78%. Sulea et al. [Sulea, Zampieri, Vela et al. (2017); Sulea, Zampieri, Malmasi et al. (2017)] applied a linear SVM classifier to predict law area and case judgments of the French Supreme Court, and reported the performance of 96%, 90% and 75.9% F1 scores in case ruling prediction, law area prediction, and estimating the time span of ruling issued, respectively. Although these efforts took full advantage of supervised learning method, they are hardly applied to other scenarios due to relying heavily on manual annotation.

Besides the judgment prediction, some researchers investigated the method of identifying applicable law articles for a given legal case. Liu et al. [Liu and Liao (2005)] proposed an intuitive solution of converting the multi-label problem into a multi-class classification problem, and obtained satisfactory results in the classification of larceny and gambling crimes. To solve the scalability issue of Liu et al. [Liu and Liao (2005)], the work in Liu et al. [Liu, Chen and Ho (2015)] reported a two-step strategy consisting of preliminary article classification by SVM and re-ranking the results using word-level features and co-occurrence tendency among articles. Luo et al. [Luo, Feng, Xu et al. (2017)] proposed an attention-based neural network to jointly implement the charge prediction and the relevant article extraction, which has reasonable generalization ability on multiple fact descriptions.

There are some works focusing on other text analysis problems. Boella et al. [Boella, Caro

and Humphreys (2011)] used TF-IDF and information gain for feature selection, and then build the SVM classifier to identify the relevant domain to which the given legal text belongs. Farzindar et al. [Farzindar and Lapalme (2004)] and Galgani et al. [Galgani, Compton and Hoffmann (2012)] studied the approach to automatic text summarization of legal documents, which can improve work efficiency of legal professionals. De Araujo et al. [De Araujo, Rigo and Barbosa (2017)] studied the problem of domain ontology-based information extraction from natural language texts, and reported an average accuracy of 96%. According to relevant law articles, sentiment analysis of crime facts and prison term, Liu et al. [Liu and Chen (2018)] use SVM algorithm to classify the judgment text automatically.

In summary, previous studies have considerably facilitated the advance in the field of artificial intelligence and law. Nevertheless, it remains a challenge to learn abundant semantic representations from case fact descriptions with less human annotations and make refined prediction of the prison term for a certain type of case. Our work in this paper aims to fill this gap.

3 Case modeling

The extensive application of NLP methods (such as word segmentation, named entity recognition, part-of-speech tagging, etc.) has remarkably advanced the processing and analysis of general textual data including news, online reviews and various social network data, and these techniques can still play a huge role in the context of legal data. However, to achieve better understanding and more effective mining of the case fact description in judgment documents, expert knowledge of relevant law articles is indispensable.

As one of the most common types of crime in judicial practice, theft cases account for over 20% of all criminal judgment documents that the Supreme People's Court of the People's Republic of China has made publicly available. Taking the theft case as the research object in this paper, we need first build its judgment model according to relevant law articles. According to Article 264 in the Criminal Law of the People's Republic of China that illustrates the basic principles and framework of judging a theft case, there are four constitutive elements of theft crime as underlined in Appendix A, which includes:

1) *Subject element*: the nature of criminal suspects that determines the criminal liability, such as age, health condition or mental status, etc.;
2) *Subjective element*: subjective intention of committing crime, and the foresight to the consequences;
3) *Object element*: the nature of articles involved in the crime, such as economic value, appropriability, mobility, etc.;
4) *Objective element*: the concealment of committing crime (to differentiate theft crime from other crimes of property violation such as the crime of forcible seizure of money or property).

A judgment document is constituted by four main parts: the *basic information about the defendant(s)*, the *fact description*, the *court's view* including relevant law articles and the judgment decision including the charge and prison term.

Through the comprehensive analysis of Article 264 and the structure of judgment document, we can describe a theft case with 11 dimensions: the value of stolen items,

whether the defendant is juvenile, whether the defendant is disabled, whether the crime can be deemed as burglary (breaking in home), whether the defendant carried lethal weapons, whether the defendant is a pickpocket, whether the crime involves other serious circumstances (including but not limited to: collision, arson, resistance to arrest, etc.), whether the defendant is a recidivist, whether the defendant returned stolen items or compensated the victim, whether the defendant voluntarily surrendered and the prison term. Specifically, the value of stolen items is the primary consideration from the perspective of judgment, the juvenile or the disabled who are convicted of theft crime may have their penalty commuted compared with ordinary people, burglary, carrying lethal weapons, pickpocket and other serious circumstances shall result in a heavier punishment, the defendant who is a recidivist shall be punished severely as well, while the behavior of surrender or compensation that can be identified as remedial measures shall contribute to obtain a mitigated punishment. Formally, the judgment model of theft cases can be expressed as:

$$C=(a, j, d, b, w, p, o, r, c, s, t) \tag{1}$$

where the description of each dimension is shown in Tab. 1.

Table 1: Description of dimensions in the judgment model of theft case

Dimension	Explanation	Range of values
a	the value of stolen items	$\{0\} \cup \mathbb{Q}_+$
j	be juvenile or not	{0, 1}
d	be disabled or not	{0, 1}
b	be a burglary or not	{0, 1}
w	carrying lethal weapons or not	{0, 1}
p	be a pickpocket or not	{0, 1}
o	existing serious circumstances or not	{0, 1}
r	be a recidivist or not	{0, 1}
c	returned/compensated or not	{0, 1}
s	surrendered or not	{0, 1}
t	prison term	$\mathbb{N}$

By integrating the structure of judgment documents with legal basis, the judgment model will facilitate an in-depth description of case details. In next section, we will describe the neural network method of feature extraction to obtain a more specific representation of the case facts, and then solve the PTP problem.

4 Method

In this section, we propose a two-step method to solve the PTP problem, as shown in Fig.

1. After the data preprocessing, the input fact description is transformed into distributed representation taking sentence as unit and fed into the sentence-level sequence encoder, and the case-level feature constructed with sentence-level feature of each dimension is then passed to train the prison term predictor.

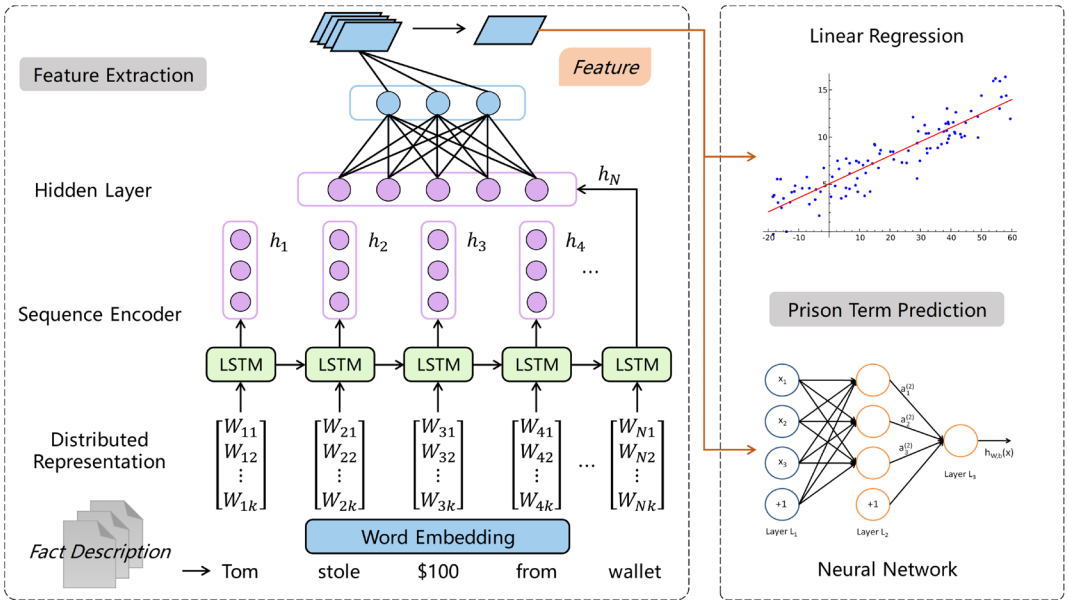

Figure 1: An overview of our method

4.1 Preliminary work

4.1.1 Text preprocessing

According to the judgment model described in Section 3, each dimension needs to be extracted from the judgment document. As all the judgment documents in our dataset are written in Chinese, the word segmentation is first carried out. After word segmentation, we remove all inessential parts from the documents except the basic information description of the defendant(s), the fact description and the judgment decision. The value of stolen items and the prison term are extracted by regular expressions[4] from the fact description part and the judgment decision part, respectively. In order to avoid the possible interference with the process of feature extraction, some insignificant words (e.g., names of people, places, organizations) are filtered by employing part-of-speech tagging and named entity recognition technology supported by the Language Technology Platform (LTP) [Che, Li and Liu (2010)].

[4] The regular expressions used to extract the value of stolen items and the prison term:
([0-9\.,，]+)[more than]?Yuan
([0-9\.,，]+)[more than]?Ten thousand[more than]?Yuan
((Defendant|Family|Relative).*(Return|Compensate|Restitute|Refund|Reimburse))|Illicit money
(Fixed-term imprisonment|Criminal detention|Public surveillance)
([1|2|3|4|5|6|7|8|9|10|11|12|13|14|15|16|17|18|19|20]+)Year(Zero?([1|2|3|4|5|6|7|8|9|10|11]+)Month)?

4.1.2 Text distributed representation

After text preprocessing, the fact description part is transformed into a word sequence. To make these Chinese words calculable, it is necessary to have each word mapped into a vector space through the distributed representation process [Mikolov, Sutskever, Chen et al. (2013)]. In this paper, we use Word2Vec and the CBOW (Continuous Bag-of-Words) model optimized by negative sampling technique to complete the distributed representation of text and map all words in the text into the same vector space.

4.2 Feature extraction

In this subsection, we aim to extract the nine-dimensional feature except the value of stolen items and the prison term from a judgment document of theft case. As each sentence in the input data has been represented as a sequence of word vectors, we can first build a sentence-level sequence encoder to embed each sentence and then merge them into the case-level feature.

RNN (Recurrent Neural Network) is a class of artificial neural network where connections between nodes form a directed graph along a sequence, which allows it to exhibit temporal dynamic behavior for a time sequence. The typical RNNs include the traditional RNN, LSTM (Long Short-Term Memory), GRU (Gated Recurrent Unit) and their variants. The ability of RNN to process variable length sequences lies in its unique neuronal structure. Taking the traditional RNN as an example, when processing the sequence information, each item of the sequence is continuously inputted into the network, and the network generates an output at each moment, then the output will jointly be processed with the input in the next moment to further generate the output in the next moment, which enables the output in each moment to carry all the information from the previous inputs. The above process can be depicted as

$$h_t=tanh\,(w_x x_t + b_x + w_h h_{t-1} + b_h) \tag{2}$$

where h_t is the output at time t, x_t is the input at time t, $h_{t\text{-}1}$ is the output at time t-1, and w and b are the parameters corresponding to x and h, respectively.

4.2.1 LSTM sequence encoder

An RNN composed of LSTM units is often called an LSTM network. LSTM is developed to deal with the exploding and vanishing gradient problems that can be encountered when training traditional RNNs. The structure of a common LSTM unit [Hochreiter and Schmidhuber (1997)] is shown in Fig. 2. It consists of a memory cell and three gates including an input gate, an output gate and a forget gate. The memory cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. Specifically, the input gate controls the extent to which a new value flows into the cell, the forget gate controls the extent to which a value remains in the cell and the output gate controls the extent to which the value in the cell is used to compute the output activation of the LSTM unit. At time step t, the forward pass of a common LSTM unit is executed as follows:

$$i_t = \sigma\,(W_i x_t + U_i h_{t-1} + b_i) \tag{3}$$

$$f_t=\sigma(W_f x_t+U_f h_{t-1}+b_f) \tag{4}$$

$$c_t=f_t\circ c_{t-1}+i_t\circ tanh(W_c x_t+U_c h_{t-1}+b_c) \tag{5}$$

$$o_t=\sigma(W_o x_t+U_o h_{t-1}+b_o) \tag{6}$$

$$h_t=o_t\circ tanh(c_t) \tag{7}$$

where matrices W_q and U_q contain the weights of the input and recurrent connections, respectively, where q can either be the input gate i, output gate o, the forget gate f or the memory cell c, depending on the activation being calculated. The operator $\circ$ is to calculate the Hadamard product of the two matrices, that is the result of multiplying the elements of the corresponding positions of the matrix.

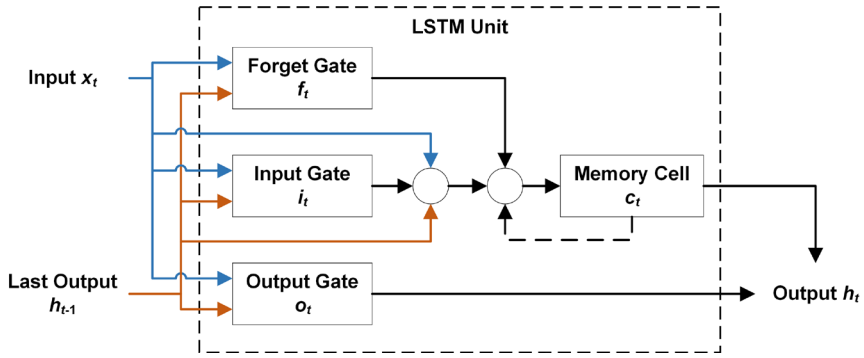

Figure 2: The structure of LSTM unit

4.2.2 GRU sequence encoder

GRU is a variant of LSTM whose unit structure [Cho, Van Merriënboer, Gülçehre et al. (2014)] is similar to LSTM but simpler, as shown in Fig. 3. Compared to LSTM, GRU removes the storage unit and the output gate, and it replaces the input gate and the forget gate with a reset gate and an update gate. At time step t, a GRU unit is updated as follows:

$$r_t=\sigma(W_r x_t+U_r h_{t-1}+b_r) \tag{8}$$

$$z_t=\sigma(W_z x_t+U_z h_{t-1}+b_z) \tag{9}$$

$$n_t=\sigma(W_n x_t+U_n(r_t\circ h_{t-1})+b_n) \tag{10}$$

$$h_t=z_t\circ h_{t-1}+(1\text{-}z_t)\circ n_t \tag{11}$$

where r_t means the result of reset gate, z_t means the result of update gate, and n_t is the intermediate result when calculating the output vector h_t.

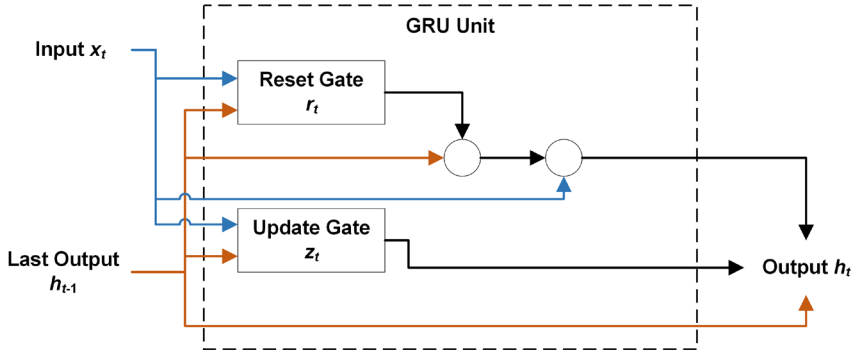

Figure 3: The structure of GRU unit

Bi-LSTM (Bi-directional LSTM) and Bi-GRU (Bi-directional GRU) are based on LSTM and GRU, respectively. They predict or label each element of the sequence based on the element's past and future contexts, by concatenating the outputs of two LSTMs or GRUs, one processing the sequence forward, the other one backward. Besides LSTM, GRU, Bi-LSTM and Bi-GRU, CNN (Convolutional Neural Network) can also be adopted to build the sequence encoder as the reference.

4.2.3 Case-level feature extraction

For a judgment document, the embedding layer in our model first transform it into a sequence of word vectors, then the sentence-level feature can be generated via the configurable sequence encoder, and the dropout layer is responsible for randomly discarding some neurons in the network to prevent over-fitting. By averaging all sentence-level features, the final case-level feature vector F_c is calculated as follows:

$$F_c = \frac{\sum_i^N F_{si}}{N} \tag{12}$$

where F_{si} means the sentence-level feature vector for the sentence i, and N is the total number of sentences.

4.3 Prison term prediction

After the process of feature extraction, we are ready to train the prison term predictor. Taking month as unit, the value of prison term is a non-negative integer, so the PTP task can be formalized as a regression problem. Here, we adopt the linear regression (LR) model and the neural network (NN) model to train the prison term predictor.

4.3.1 LR predictor

LR is a linear approach to modelling the relationship between a dependent variable and one or more independent variables. If there is only one independent variable, the process is called simple linear regression, while for more than one independent variable, it is called multiple linear regression.

For the PTP problem, there are 9 independent variables, so it is a multiple linear regression problem. The model takes the form as follows:

$$y=\beta_0 \mathbf{1}+\beta_1 x_1+\cdots+\beta_p x_p+\varepsilon=x^T\beta+\varepsilon \quad (13)$$

In order to get the linear relationship between the dependent variable y and the p-vector of regressors x, the least-squares estimation is used to fit the linear regression model.

4.3.2 NN predictor

The NN is suitable for dealing with nonlinear problems. As there can be multiple dependent variables and independent variables, it is often used for multi-label classification. It is also feasible to employ NN to solve the regression problem by removing the activation function, setting one node in the output layer, and changing the loss function to the mean square error.

5 Experiments

5.1 Dataset

We collect and construct a real-world dataset containing 41,481 judgment documents of theft cases published by China Judgments Online[5]. The dataset covers 527 grass-roots courts in eight provinces and cities, as shown in Fig. 4.

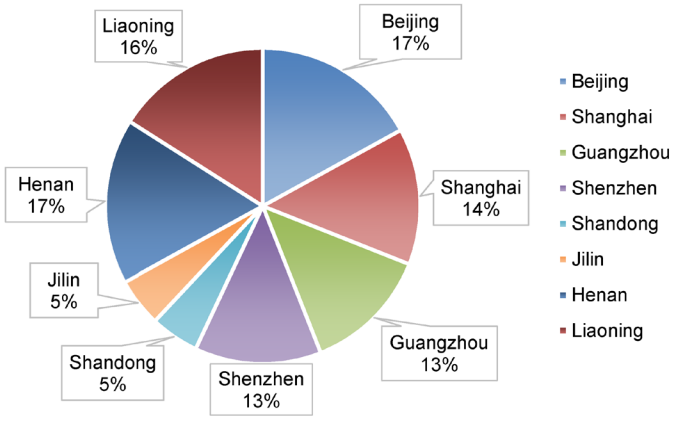

Figure 4: Dataset distribution in 8 provinces and cities

5.2 Feature extraction

To evaluate the performance of our feature extraction method, we first manually annotate 7,079 judgment documents as the training data. And to test the effect of word vector dimension on the performance of feature extraction, the word vectors are trained in the dimensions of 100, 150, 200, 250 and 300, respectively, after the text preprocessing. Then we perform a 10-fold cross-validation on the training set with LSTM, GRU, Bi-LSTM, Bi-GRU and CNN, respectively.

Tab. 2 shows the results of feature extraction with different neural network sequence encoders and word vector dimensions. Fig. 5 provides a more intuitive view about the

[5] http://wenshu.court.gov.cn/

performance difference among the five neural network sequence encoders, from which we can observe that GRU slightly exceeds other models. From the perspective of word vector dimension, CNN and LSTM obtain the highest accuracy of 98.27% and 99.23% with 150-dimension word vector, GRU and Bi-LSTM obtain the highest accuracy of 99.45% and 99.12% with 300-dimension word vector, and Bi-GRU obtains the highest accuracy of 99.23% with 250-dimension word vector. The highest accuracy of feature extraction is achieved using GRU sequence encoder with a 300-dimension word vector, so we use this setting in the following evaluation.

Table 2: Accuracy (%) of feature extraction

Model \ Dimension	100	150	200	250	300
CNN	97.75	**98.27**	97.75	96.00	96.02
LSTM	97.65	**99.23**	98.56	98.38	98.91
GRU	99.02	99.35	99.24	99.37	**99.45**
Bi-LSTM	93.93	98.45	98.93	99.04	**99.12**
Bi-GRU	98.75	99.04	99.14	**99.26**	99.20

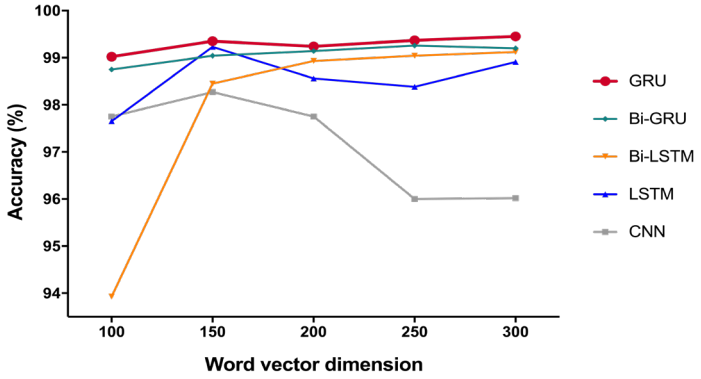

Figure 5: Accuracy of feature extraction

5.3 Prison term prediction

In this subsection, we aim to evaluate our method of prison term prediction from two perspectives: the prediction model, and the dataset.

5.3.1 Performance of different prediction models

Our method is characterized by the incorporation of neural network predictor and feature extraction based on judgment model. To evaluate the effectiveness of the judgment model, we need to build contrast predictors that simply use word vectors as the text feature without legal basis. Here, we adopt LSTM, Bi-LSTM, GRU and Bi-GRU models, respectively, to train the word vectors and make the prediction. They are trained over the

same dataset as GRU+LR and GRU+NN predictors we proposed, among which 60% are for training, 20% each for validation and testing.

We employ three indicators for evaluation metrics, which are: 1) **MAE** (lower is better): mean absolute error of prison term between the predicted numbers of months versus observed, 2) **Acc_e3** (higher is better): the percentage of predicted results with errors not more than three months (i.e. the error upper bound is three months), and 3) **Acc_e6** (higher is better): the percentage of predicted results with errors not more than six months (i.e., the error upper bound is six months).

Results are shown in Tab. 3, from which we can infer that both GRU+LR and GRU+NN predictors consistently and significantly outperform all the contrast models, and GRU+NN obtains the best performance of 3.2087 months in MAE, 72.54% in Acc_e3, and 90.01% in Acc_e6, respectively. The experimental results demonstrate the effectiveness of our method which empowers the feature extraction with judgment model and legal basis.

Table 3: Performance of different prediction models

Model	MAE (months)	Acc_e3 (%)	Acc_e6 (%)
Bi-LSTM	4.3727	50.86	84.14
LSTM	4.1705	55.53	86.02
Bi-GRU	4.1635	58.00	86.27
GRU	4.0003	63.09	86.49
GRU+LR	3.3896	71.13	89.95
GRU+NN	**3.2087**	**72.54**	**90.01**

5.3.2 Performance over different datasets

It is intuitive that the judicial decision may be affected by some factors varying among different courts or regions. In this group of experiments, we further explore the effect of dataset on the performance of our method. We divide the universal set containing 41,481 judgment documents into 8 subsets by regions shown in Fig. 4, and then retrain and evaluate the GRU+LR predictor and GRU+NN predictor over the 8 subsets, respectively.

Tab. 4 shows the comparison of PTP results among the universal set and 8 subsets. We have the following observations: 1) our model obtains a relatively better accuracy over Guangzhou and Shenzhen subsets than others even the universal, 2) the performance drops considerably over Shandong and Jilin subsets account for two least proportions of universal sets. It demonstrates that a large-scale dataset would in general facilitate the understanding of legal texts and benefit the training of prediction model. But it should be noted that the format of judgment documents may vary among different regions, which will lead to the inaccuracy of feature extraction, this is why the growth of dataset scale does not always boost the performance of our method.

Table 4: Effect of dataset on performance of our method

Dataset	MAE (months)	Acc_e3 (%)	Acc_e6 (%)
	(GRU+LR Predictor/GRU+NN Predictor)		
Universal Set	3.3896/3.2087	71.13/72.54	89.95/90.01
Beijing	2.9206/2.8264	77.83/78.61	93.62/93.76
Shanghai	3.6389/3.4231	73.23/76.72	90.39/90.31
Guangzhou	2.5407/2.6481	81.11/77.88	92.03/92.92
Shenzhen	2.8711/2.8095	81.69/82.25	92.34/92.25
Shandong	4.0323/3.6169	63.68/65.92	84.07/84.82
Jilin	5.1015/4.7864	51.30/56.77	81.51/84.63
Henan	3.5528/3.3129	66.52/70.02	86.79/88.47
Liaoning	3.7009/3.4358	63.25/68.10	88.62/89.23

6 Conclusion

In this paper, we investigated an approach to prison term prediction on criminal case description. To obtain a better understanding and more specific representation of the legal texts, we summarized the judgment model of theft cases according to relevant law articles. Several state-of-the-art neural networks were employed to implement the extraction of judgment-specific case feature. We adopted the linear regression model and the neural network model to build the prison term predictor. Experimental results on the real-world dataset demonstrated the effectiveness of our method.

In future work, we will expand the dataset, and further validate and advance the proposed method on various types of criminal case.

Acknowledgement: This work is supported in part by the National Key Research and Development Program of China under grants 2018YFC0830602 and 2016QY03D0501, and in part by the National Natural Science Foundation of China (NSFC) under grants 61872111, 61732022 and 61601146. We thank the reviewers for their constructive suggestions to improve the quality of the paper.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Aletras, N.; Tsarapatsanis, D.; Preoţiuc-Pietro, D.; Lampos, V. (2016): Predicting judicial decisions of the European Court of Human Rights: a natural language processing perspective. *PeerJ Computer Science*.

Boella, G.; Di Caro, L.; Humphreys, L. (2011): Using classification to support legal knowledge engineers in the Eunomos legal document management system. *Proceedings of the Fifth International Workshop on Juris-Informatics*.

Che, W.; Li, Z.; Liu, T. (2010): LTP: A Chinese language technology platform. *Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations*, pp. 13-16.

Cho, K.; van Merriënboer, B.; Gülçehre, C.; Bahdanau, D.; Bougares, F. et al. (2014): Learning phrase representations using RNN encoder-decoder for statistical machine translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1724-1734.

De Araujo, D. A.; Rigo, S. J.; Barbosa, J. L. V. (2017): Ontology-based information extraction for juridical events with case studies in Brazilian legal realm. *Artificial Intelligence and Law*, vol. 25, no. 4, pp. 379-396.

Farzindar, A.; Lapalme, G. (2004): Legal text summarization by exploration of the thematic structure and argumentative roles. *Proceedings of the Text Summarization Branches Out Workshop*, pp. 27-38.

Galgani, F.; Compton, P.; Hoffmann, A. (2012): Combining different summarization techniques for legal text. *Proceedings of the Workshop on Innovative Hybrid Approaches to the Processing of Textual Data*, pp. 115-123.

Gonçalves, T.; Quaresma, P. (2005): Evaluating preprocessing techniques in a text classification problem. *Proceedings of the Conference of Brazilian Computer Society*.

Hachey, B.; Grover, C. (2006): Extractive summarisation of legal texts. *Artificial Intelligence and Law*, vol. 14, no. 4, pp. 305-345.

Hochreiter, S.; Schmidhuber, J. (1997): Long short-term memory. *Neural computation*, vol. 9, no. 8, pp. 1735-1780.

Katz, D. M.; Bommarito II, M. J.; Blackman, J. (2017): A general approach for predicting the behavior of the Supreme Court of the United States. *PLoS One*, vol. 12, no. 4, e0174698.

Lin, W. C.; Kuo, T. T.; Chang, T. J.; Yen, C. A.; Chen, C. J. et al. (2012): Exploiting machine learning models for Chinese legal documents labeling, case classification, and sentencing prediction. *Computational Linguistics and Chinese Language Processing*, vol. 17, no. 4, pp. 49-68.

Liu, C. L.; Chang, C. T.; Ho, J. H. (2004): Case instance generation and refinement for case-based criminal summary judgments in Chinese. *Journal of Information Science and Engineering*, vol. 20, no. 4, pp. 783-800.

Liu, C. L.; Hsieh, C. D. (2006): Exploring phrase-based classification of judicial documents for criminal charges in Chinese. *Proceedings of the 16th International Symposium on Methodologies for Intelligent Systems*, pp. 681-690.

Liu, C. L.; Liao, T. M. (2005): Classifying criminal charges in Chinese for web-based legal services. *Proceedings of the 7th Asia-Pacific Web Conference*, pp. 64-75.

Liu, Y. H.; Chen, Y. L. (2018): A two-phase sentiment analysis approach for judgement prediction. *Journal of Information Science*, vol. 44, no. 5, pp. 594-607.

Liu, Y. H.; Chen, Y. L.; Ho, W. L. (2015): Predicting associated statutes for legal problems. *Information Processing & Management*, vol. 51, no. 1, pp. 194-211.

Luo, B.; Feng, Y.; Xu, J.; Zhang, X.; Zhao, D. (2017): Learning to predict charges for criminal cases with legal basis. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2727-2736.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; Dean, J. (2013): Distributed representations of words and phrases and their compositionality. *Proceedings of the 27th Conference on Neural Information Processing Systems*, pp. 3111-3119.

Palau, R. M.; Moens, M. F. (2009): Argumentation mining: the detection, classification and structure of arguments in text. *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, pp. 98-107.

Sulea, O. M.; Zampieri, M.; Malmasi, S.; Vela, M.; Dinu, L. P. et al. (2017): Exploring the use of text classification in the legal domain. *Proceedings of the Second Workshop on Automated Semantic Analysis of Information in Legal Texts co-located with the 16th International Conference on Artificial Intelligence and Law*.

Sulea, O. M.; Zampieri, M.; Vela, M.; van Genabith, J. (2017): Predicting the law area and decisions of French Supreme Court cases. *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, pp. 716-722.

Xiong, Z.; Shen, Q.; Wang, Y.; Zhu, C. (2018): Paragraph vector representation based on word to vector and CNN learning. *Computers, Materials & Continua*, vol. 55, no. 2, pp. 213-227.

Appendix A. Article 264 of the Criminal Law of the P.R. China

In accordance with the Amendment VIII to the Criminal Law of the People's Republic of China promulgated on February 25th, 2011, this Article is amended to read:

"*Whoever steals a relatively large amount of public or private property, or commits theft repeatedly, or commits burglary, or steals with a lethal weapon, or pickpockets, shall be sentenced to fixed-term imprisonment of not more than three years, criminal detention or public surveillance and be concurrently or separately fined. If the amount is huge or there are other grave circumstances, he shall be sentenced to fixed-term imprisonment of not less than three years but not more than ten years and be concurrently fined. If the amount is especially huge or there are other especially serious circumstances, he shall be sentenced to fixed-term imprisonment of not less than ten years or life imprisonment, and be concurrently subject to a fine or confiscation of property.*"