A Judicial Sentencing Method Based on Fused Deep Neural Networks

Yuhan Yin, Hongtian Yang, Zhihong Zhao, and Songyu Chen

Nanjing University, Nanjing, China
{yinyuhan,mgl632009}@smail.nju.edu.cn

Abstract. Nowadays, the judicial system has been hard to satisfy the growing judicial needs of the people. Therefore, the introduction of artificial intelligence into the judicial field is an inevitable trend. This paper incorporates deep learning into intelligent judicial sentencing and proposes a comprehensive network fusion model based on massive legal documents. The proposed method combines multiple networks, e.g., recurrent neural network and convolutional neural network, in the procedure of sentencing prediction. Specially, we use text classification and post-classification regression to predict the defendant's conviction, articles of law related to the case and prison term. Moreover, we use the simulated gradient descent method to build a fusion model. Experimental results on legal documents datasets justify the effectiveness of the proposed method in sentencing prediction. The fused network model outperforms each individual model in terms of higher accuracy and stability when predicting the conviction, law article and prison term.

Keywords: Intelligent judiciary · Text classification · Neural network

1 Introduction

With the development of society and the improvement of the judicial ability, the demands of people for judicial services are increasing, and the problems of traditional judicature are increasingly serious. First of all, knowledge reserves and business level of judicial official cannot satisfy social need because of a numerous number of laws and legal theories. In addition, the judicial standard fails to unify and get a fair state since judicial personals have uncertain knowledge reserves. For cumbersome legal provisions and case details, legal practitioners need to do a lot of repetitive work to judge sentencing. These problems have pushed artificial intelligence to the frontline in judicial field [1, 2]. In order to promote the intellectualization of the judicial field, we proposed a judicial sentencing method based on fused deep neural networks. In this method, the prediction of conviction and law article can be abstracted as text classification problems in natural language processing, while the defendant's prison term prediction can be formulated into a regression problem based on text classification. With the continuous improvement of neural network model, it is of great importance to develop a more effective model specifically for such issues to make the judiciary more intelligent.

I. V. Tetko et al. (Eds.): ICANN 2019, LNCS 11730, pp. 213–226, 2019.
https://doi.org/10.1007/978-3-030-30490-4_18

Current research directions include using intelligent lie detectors to assist trials, biotechnical suspect tracking and research on accessorial intelligent sentencing which is to be discussed in this paper. In [3] the authors mentioned that Britain proposed using an evidence-based accessorial system in the prosecution work of the Crown Prosecutor's Office. In 2005, Gao started to take professional research on intelligent sentencing with an emphasis on the crime of Larceny. Based on a massive number of cases, she used support vector machine (SVM) to predict the sentence of the defendant. Intelligent sentencing prediction can give out objective answers on sentencing rationally by using only legal knowledge and allow more people to conduct judicial supervision without understanding the criminal law, which is crucial for constructing a fair social environment.

Natural language processing (NLP) techniques are widely-used in intelligent sentencing prediction. In general, the processing of natural language text classification can be decomposed into four steps, including text preprocessing, text representation, text feature extraction, and text classification model construction. The purpose of text representation is to represent the pre-processed text in the way that computer can understand. The traditional approach is Bag of Words (BoW) [4]. The main shortcoming of this model is that it does not take text context into account. In recent years, Mikolov et al. [5] proposed the Word2Vec model for word embedding. Then, feature extraction mainly includes methods such as word frequency [6], document frequency and Information Gain [7]. In the subsequent step, we need to select the appropriate classifier to train the selected features. Traditional classifiers include SVM [8], nearest neighbor classification, Bayesian classification, decision tree, RBF neural networks, and random forest etc. With the emergence of neural networks, researchers have started to use recurrent neural networks (RNN) for text classification [9]. Kim [10] has applied word embedding to the simple CNN structure to construct a classic text classification model. Mikolov [11] proposed the FastText model, which is a simple and fast model based on BoW. Therefore, on the basis of deep learning, it is an inevitable trend to use the existing neural networks for intelligent judicial sentencing research.

Relying on multiple neural networks, this paper proposes a NLP-based judicial sentencing method. It fuses neural networks such as RNN, CNN, RCNN and FastText to perform sentencing prediction. Specifically, the technical contributions of this work can be summarized as follows: (1) In the issue of judicial sentencing, we compare the performance of a number of different neural networks in prediction. (2) Compared to the ordinary regression model, we proposed the idea of regression after classification can significantly improve the accuracy of regression. (3) We use the simulated gradient descent method to build a fusion model, and the model improves the performance of sentencing prediction. Experimental results show that the accuracies in predicting conviction, law article and prison term by the proposed method are 92%, 91% and 74%, respectively.

The rest of this paper is organized as follows. Section 2 introduces the preliminary knowledge relevant to this work. Section 3 details the model structure and explains the proposed method that combines multiple neural networks. Experimental results are demonstrated in Sect. 4. Finally, Sect. 5 concludes the whole paper and discusses future work directions.

2 Preliminaries

Deep learning is of great significance in natural language processing tasks. In recent years, neural network algorithms have also been frequently applied in text categorization. This section introduces TextCNN, TextRNN and TextRCNN, which will be used in this paper.

TextCNN is an algorithm for text categorization based on convolutional neural networks. It was proposed by Yoon Kim in 2014. Figure 1 shows the structure and mechanism for TextCNN. First of all, Kim convolves the text matrix with filters of different lengths, where the width of the filter is equal to the length of the word embedding. Then he use the max-pooling layer to operate on the vector generated by every filter so that each filter only produces a maximum value of the region. Next, the maximum values generated by different filters are spliced to compose an abstract vector that characterizes the sentence. Finally, the prediction is made relying on this vector. This model is one of the classic models in natural language processing.

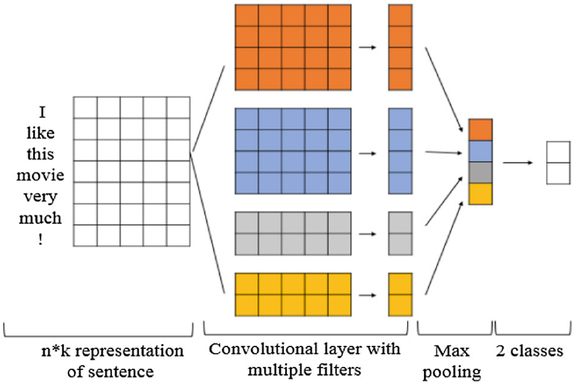

I
like
this
movie
very
much
!
n*k representation of sentence
Convolutional layer with multiple filters
Max pooling
2 classes

Fig. 1. TextCNN structure. (adapted from [10])

However, a significant disadvantage of TextCNN is that the convolution and pooling operations lose the order and position information of the words in the text sequence, and it is difficult to capture the semantic information such as negation and antisense in the text sequence. In contrast, RNN can capture sequence information and RNN is a general-purpose model for modeling sequences [12]. Therefore, TextRNN is also one of the networks that NLP scholars are keen on.

In recent years, Bidirectional RNN (Bi-RNN) [13] is used more and more frequently due to its capability of taking more context information into account. The structure of the Bi-RNN is the combination of two unidirectional RNNs. At each moment, there are two RNNs in opposite directions in the input layer, and the output layer is determined by this two unidirectional RNNs. Figure 2 is a Bi-RNN structure expanding over time. In Bi-RNN, there are six unique weights being reused at each moment, including w1, w3 used from the input layer to the forward hidden layer and to the backward hidden layer, w2 and w5 used from one hidden layer to another hidden layer and w4, w6 used from the forward hidden layer and backward hidden layer to the output layer.

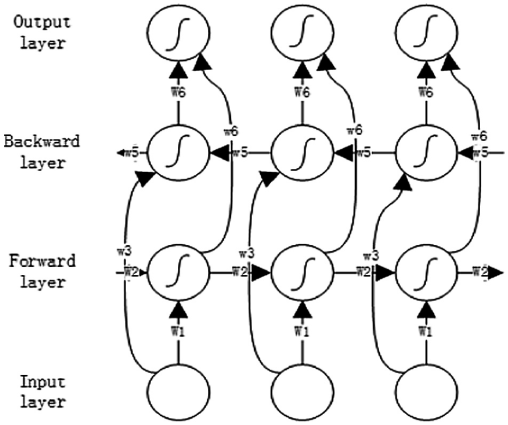


Fig. 2. Bi-RNN structure diagram. (adapted from [13])

There are two common variants of RNN: Gated Recurrent Unit (GRU) [14] and Long Short-Term Memory (LSTM) [15]. GRU has only two gates (update and reset), and it directly passes the hidden state to the next unit. Meanwhile, LSTM has three gates (forget, input, output), and the hidden state is wrapped with the memory cell. GRU and LSTM, either of which is a variant of RNN, have compatible performance in many tasks, and they both perform better than standard RNN. However, due to the difference in their structure, the GRU is less likely to converge because of its fewer parameters, which means the GRU is suitable for the case with less data volume, while the LSTM has better ‘memory’ effect and better performance when the training data set is large enough.

In 2015, the Chinese Academy of Sciences proposed a combined model of RNN and CNN [16], called RCNN. The RCNN model is one of the models commonly used in recent years. It is based on the two models above, and Fig. 3 shows its structure.

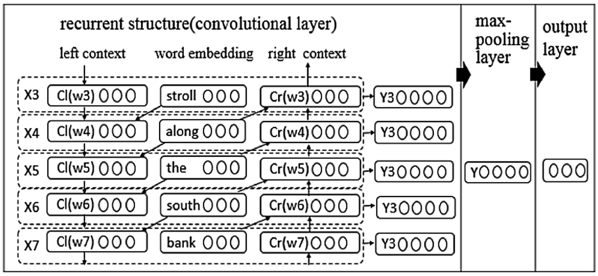


Fig. 3. RCNN structure diagram. (adapted from [16])

In this model, the word embedding of a word is not only the word embedding, but also the vector splicing of the word and its both side neighbors. Therefore, Bi-LSTM will be used for word vectorization before the CNN since it can remarkably improve the relevance of words and the accuracy of text vectors. Then, the constructed Bi-LSTM is connected to the TextCNN, the filter size of which is set as 1, and the rest of the parameters stay unchanged.

3 The Proposed Methods

Judicial sentencing prediction is the procedure of making predictions on the defendant's conviction of crime (task 1), articles of law concerned with the case (task 2) and prison term (task 3) via text analysis on legal documents including the case descriptions and factual statements.

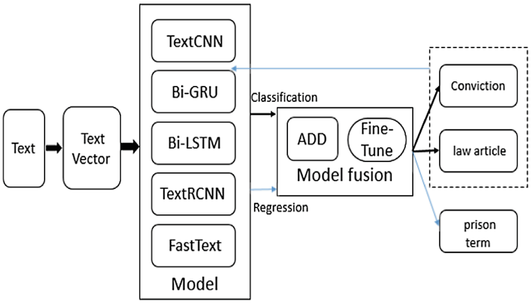


Fig. 4. The intelligent sentencing model structure diagram.

In this paper, the intelligent sentencing model based on fused neural networks is illustrated in Fig. 4. The proposed model aims to predict the conviction and the related law articles in the first place, and then predicts the prison term. For different model results, we propose a fusion method to make model weights adjustments. The final result can be obtained by weighted averaging.

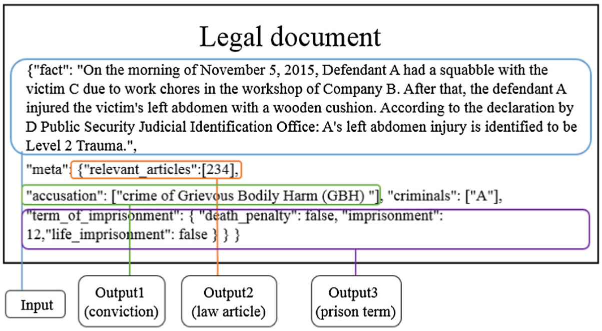
3.1 Data Analysis and Preprocessing

In this section, we analyzes the text data of the legal documents in the training set, conducts targeted preprocessing work and then take these processed data as the input to the next stage in the model training.

Data Analysis

In this section, we analyzes the text data of the legal documents in the training set, conducts targeted preprocessing work and then take these processed data as the input to the next stage in the model training.

In this work, the contents of the legal documents and the input and output format of the model are shown in Fig. 5.



```
{"fact": "On the morning of November 5, 2015, Defendant A had a squabble with the victim C due to work chores in the workshop of Company B. After that, the defendant A injured the victim's left abdomen with a wooden cushion. According to the declaration by D Public Security Judicial Identification Office: A's left abdomen injury is identified to be Level 2 Trauma.",
"meta": {"relevant_articles":[234],
"accusation": ["crime of Grievous Bodily Harm (GBH) "], "criminals": ["A"],
"term_of_imprisonment": { "death_penalty": false, "imprisonment": 12,"life_imprisonment": false } } }
```

Fig. 5. An example of legal document structure and the Input/Output of the model.

Firstly, we observe the distribution of legal documents, two messages are conveyed on our research: (a) The distribution of convictions and law articles is not uniform, thus it is difficult to make predictions on rare samples. (b) There is an obvious co-occurrence phenomenon between convictions and articles of law.

Then, we analyze the two messages above. For message (a), since the sample data are extremely unbalanced in composition, resampling is necessary before data processing. For message (b), the co-occurrence phenomenon proves that convictions and law articles have the same rules and can be predicted in the same way. It is commonly known that prison term is closely related to the conviction and law articles. So, we prefer the following information topology among the three tasks (Fig. 6).

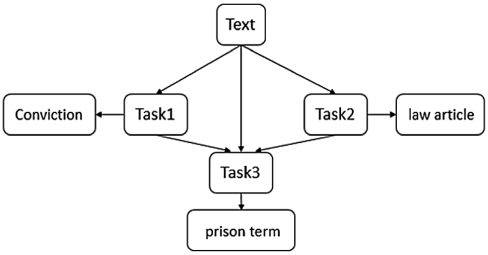


Fig. 6. Topology diagram among three tasks.

Data Processing

This section mainly describes the pre-processing procedure of legal document, and the main steps are as follows.

(a) Resampling. The data of rare samples are made multiple copies of so that model can achieve a more balanced composition of data.

(b) Data extraction. The description text, convictions, articles and prison terms which are the input and output of three tasks, are extracted from the legal documents. At the same time, convictions are converted into indexed values in the list of charges.

(c) Word segmentation [17]: We use Jieba (the Chinese word segmentation tool) for word segmentation. In this step, it is necessary to delete common punctuation symbols because legal documents are supposed to be objective and non-emotional.

(d) Word embedding training: In order to achieve semantic analysis, Word2vec is applied to train word vector and get the corresponding word features of the legal documents. Word2Vec model mainly includes the Continuous Bag-of-Words Mode (CBOW) and the Skip-Gram model. This article uses the Skip-Gram model to represent words as word embedding after word segmentation.

(e) Text truncation: In this experiment, text with different lengths needs to be converted into fixed-length text segment. That is to say, a text length N is set, and any text longer than N should be truncated while text shorted than N will be completed. We analyze the data and know that the length of legal documents mostly is between 200 and 500. Then, the experiments of N from 200 to 500 show that N is the best when it is 350. Because of the particularity of legal documents that the second half contains too much key information, in this experimental we will cut the text from back to front into 350 dimensions.

(f) Text data enhancement processing: In the construction of neural network models, the training data greatly affects the prediction results. In order to make the prediction results more accurate, data enhancement processing is necessary, including shuffle and dropping. This processing will reduce the over-fitting of the model and make the model training process more balanced. Data enhancement is applied as the final step before training to avoid the possible consequence that some deleted data details were not found in the previous steps.

3.2 Models

In this section, the pre-processed text vectors are trained by different deep neural networks that includes CNN, RCNN, Bi-LSTM and Bi-GRU respectively. By continuously adjusting the parameters of the neural network model, the three tasks obtained good experimental results. The parameters and structure are as follows (Fig. 7).

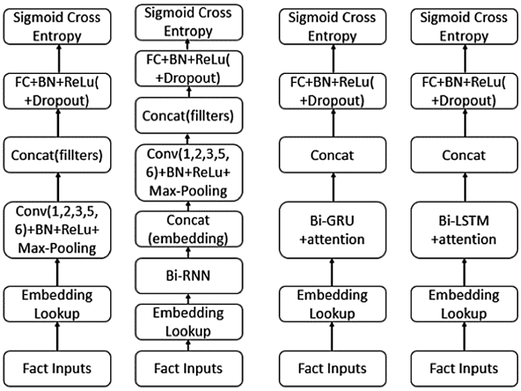


Fig. 7. The parameters and structure of the four neural network models

3.3 Model Fusion

Model fusion is one of the important steps in machine learning. Model fusion not only makes prediction results more accurate, but also can reduce the disadvantages of individual models. This experiment uses a weighted adjustment method that simulates the gradient descent method. Firstly, it is necessary to construct a prediction probability matrix of a singles model, in which the probability matrix in the single model *i* is P_i. Then the weight W_i is assigned according to the performance of the single model *i*. Finally, the model calculates the F1 value of the fusion verification set. Among them, score is a function to measure the performance of the model at this time.

In this experiment, CNN, LSTM, RCNN models with good experimental results and FastText with poor experimental results were selected for model fusion.

Algorithm 1: Model Fusion

```
Input:
 weights, //Array of model weights
 label_list, //a list of output matrices of different models
 label,  //Correct output matrix of training set
 lr,   //Learning rate
 k //The number of iterations
Output:
 The highest model fusion accuracy F1,
 The weights of the highest accuracy
calculate the F1 of initial weights;
for i = 0 to k do
  for j = 0 to len(weights) do
    weights[j] <- weights[j] + lr;
    calculate score(weights);
    if score(weights) > F1 then
      calculate weights[j];
    else weights[j] <- weights[j] - lr;
      calculate score(weights);
      if score(weights) > F1 then
        calculate weights[j];
    endif
  endfor
  update weights;
  if (score(new weights) > F1) then
     F1 <- score(new weights);
     record weights;
endfor
return F1,weights;
```

$$P = \sum\nolimits_{1}^{k} w_i P_i \tag{1}$$

$$F1_{last} = score(P) \tag{2}$$

Algorithm 1 describes the algorithmic flow of the proposed network fusion model. Corresponding to lines 3–13, add and subtract the learning rate for the weight of each individual model, then update the weight if F1 is raised. In line 15–17, update the largest F1 and corresponding weight array after each round. After setting the different learning rate and the initial weight, the model fusion process ends with the F1 value stable through multiple iterations.

4 Experimental Results

4.1 Data Set

In this paper, the data set contains a set of legal documents, including 202 convictions, 183 law articles and prison term [18]. Prison term is divided into death penalty, life imprisonment and 0–300 months of imprisonment. This experiment selects 20,000 legal documents as test documents.

4.2 Performance Evaluation

This paper covers the evaluation on multiple classification problems and regression problems. For classification problems, performance evaluation is required through the confusion matrix. Specifically for the two classification problem, sample data can be classified into four categories according to the correct category and the category that predicted by classifier: true positive (TP), false positive (FP), false negative (FN), and true negative (TN). Precision (P), Recall (R) and F1 are used to test the classification accuracy [19].

$$P = \frac{TP}{TP + FP}, R = \frac{TP}{TP + FN}, F1 = \frac{2 * P * R}{P + R} \tag{3}$$

Macro-averaging (Macro-P) and micro-averaging (Micro-P) [20] need to be introduced in the case of N classification.

$$\begin{gathered} Macro_p = \tfrac{1}{n}\sum_{i=1}^{n} P_i, Macro_R = \tfrac{1}{n}\sum_{i=1}^{n} R_i, \\ Macro_F = \frac{2 * Macro_P * Macro_R}{Macro_P + Macro_R} \end{gathered} \tag{4}$$

$$\begin{gathered} Micro_P = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FP_i}, Micro_R = \frac{\sum_{i=1}^{n} TP_i}{\sum_{i=1}^{n} TP_i + \sum_{i=1}^{n} FN_i} \\ Micro_F = \frac{2 * Micro_P * Micro_R}{Micro_P + Micro_R} \end{gathered} \tag{5}$$

The evaluation methods of conviction prediction and law article prediction in this paper are as follows:

$$Score_{1,2} = \frac{Macro_F + Micro_F}{2} \tag{6}$$

In the prediction of prison term, if the result of the prediction is V1, the actual term is V2:

$$V = 1 - abs(\log(V1 + 1) - \log(V2 + 1)) \tag{7}$$

$$Score_3 = \frac{\sum_{i=1}^{n} V_i}{N} \tag{8}$$

Score1, 2, 3 represent the accuracy of task 1, 2, 3 respectively. The higher value of score means the classification performs better.

4.3 Experimental Results and Analysis

In this experiment, it is necessary to determine the length of text truncation input into the model before everything else. Taking CNN model as an example, the effects of different text lengths on the performance of the model are shown below. Therefore, 350 dimension is chosen as the most suitable truncation length in this paper (Fig. 8).

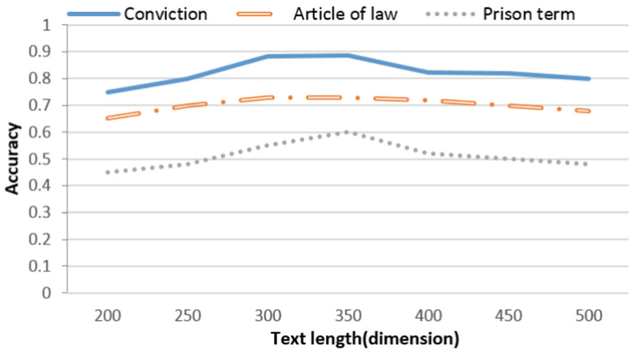


Fig. 8. The Relation of text length and prediction accuracy.

In the experiment, we use the deep neural networks LSTM, GRU, RCNN and CNN to predict the three tasks firstly. In addition, the comparative experiments of FastText and Support Vector Machine (SVM) are carried out. According to the experimental results in the Table 1, the performance based on deep neural networks CNN, RNN and RCNN is better than that of shallow networks like FastText and SVM. SVM has the worst prediction effect because of data imbalance and big data set. Finally, the fusion model can significantly improve the accuracy of sentencing prediction.

Each neural network model has its pros and cons. In order to improve the accuracy of prediction, this paper chooses several networks for model fusion. LSTM, GRU, RCNN and CNN are primary choices in this experiment. FastText tends to have a poor prediction result since its hidden layer is obtained by simply averaging the sum, but this shortcoming can be made up for by RCNN, whose focus on contextual connections. Therefore, FastText is also included in the fusion.

Table 1. Comparison of prediction accuracies.

Model	Conviction (Score1)	Law articles (Score2)	Prison term (Score3)
LSTM	0.90	0.89	0.71
GRU	0.89	0.88	0.70
RCNN	0.88	0.87	0.70
CNN	0.87	0.87	0.69
FastText	0.78	0.78	0.58
SVM	0.71	0.69	0.48
Fusion model	0.92	0.91	0.74

Since the model fusion adopts the simulated gradient descent method, it is possible for model to fall into local optimum. In order to handle this situation, multiple weight adjustments are needed to find the optimal results. After sufficient amounts of experiments, the optimal weights and performance evaluation are found out as follows.

Table 2. Weigh distribution and the score of conviction.

LSTM	GRU	RCNN	CNN	FastText	Score1
1.03	0.78	0.65	−0.53	−0.93	0.92

Table 3. Weigh distribution and the score of law article.

LSTM	GRU	RCNN	CNN	FastText	Score2
0.53	0.33	0.25	0.18	−0.29	0.91

Table 4. Weigh distribution and the score of prison term.

LSTM	GRU	RCNN	CNN	FastText	Score3
0.95	0.69	0.30	−0.06	−0.88	0.74

In Tables 2, 3 and 4, the weights of the fusion model and the experimental results justify that models with good performance should be assigned with large weights, while models with bad performance should be assigned with small weights. Furthermore, if the prediction result of a specific model has a positive impact on the final performance of the fusion model, this model should be assigned a positive weight. On the contrary, a model that has a negative impact should be assigned a negative weight.

Finally, the prediction performance can be improved by 2% to 3% through model fusion, and the global accuracy can reach 92%, 91% and 74%.

Table 5. The top three of the error rate in task 1

Conviction	Crime of scalping relics	Crime of ill-treatment of prisoner	Crime of hijacking a ship or car
Error rate	0.45	0.30	0.25

Having verified the prediction accuracy of the fusion model, it is necessary to further perform an error analysis on the frequency of large deviations. Take task 1 as an example, because the distribution of crimes is uneven, it is unreasonable to compare the error frequency of all categories. In this paper, we mainly analyze the error rate of a category which is the proportion of the number of category's prediction errors to the total quantity of the category. After summarizing the conviction categories of predicting the wrong result, there is a common regularity in convictions with higher error rates is the number of those in the training set does not exceed 500. Among them, the top three in the error rate ranking are crime of scalping relics, crime of ill-treatment of prisoner and crime of hijacking a ship or car. Table 5 lists the error rates of these three categories. We can conclude accordingly that, the categories with high prediction error rates are those categories that have an insufficient number of samples in the train set. It can be expected that the prediction accuracy can be significantly improved given sufficient number of samples.

5 Conclusions and Future Work

This paper introduces deep learning into the field of judicial sentencing, and abstracts intelligent sentencing into text classification in natural language processing and regression after classification. Through the training of CNN, RNN, RCNN, FastText and SVM, the performance of different models is compared in intelligent sentencing. Finally, it is meaningful to analyze and select the corresponding model for fusion to get a better prediction model. In our future research, the experiments will capture the main components in law documents, and then strengthen the useful text fragments while weakening the useless text fragments to highlight the text features. Moreover, we consider adding legal knowledge to further improve the prediction accuracy, making more significant contributions to the study of judicial sentencing based on deep learning. Last but not least, it is essential to improve the accuracy of categories that have an insufficient number of samples in the train set.

Acknowledgment. This paper is supported by the National Key Research and Development Program of China (Grant No. 2018YFC1801605), the Fund of State Key Laboratory for Novel Software Technology at Nanjing University (No. ZZKT2018B01).

References

1. Aletras, N., Tsarapatsanis, D., Preoţiuc-Pietro, D., Lampos, V.: Predicting judicial decisions of the European Court of Human Rights: a natural language processing perspective. PeerJ Comput. Sci. **24**(2), e93 (2016). https://doi.org/10.7717/peerj-cs.93
2. Schild, U.J.: Criminal sentencing and intelligent decision support. In: Sartor, G., Branting, K. (eds.) Judicial Applications of Artificial Intelligence, pp. 47–98. Springer, Dordrecht (1998). https://doi.org/10.1007/978-94-015-9010-5_3
3. Zong, B.: On the application of artificial intelligence in the judgment of criminal proof standard. Sci. Law (J. Northwest Univ. Polit. Sci. Law). https://doi.org/10.16290/j.cnki.1674-5205.2019.01.004
4. Kantor, P.: Foundations of statistical natural language processing. Inf. Retrieval **4**(1), 80–81 (2001). https://doi.org/10.1023/A:1011424425034
5. Mikolov, T., Sutskever, I., Chen, K., Dean, J., Corrado, G.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems (2013)
6. Eszter, B., István, C., Dániel, K., et al.: Race, religion and the city: twitter word frequency patterns reveal dominant demographic dimensions in the United States. Social Science Electronic Publishing (2016). https://doi.org/10.1057/palcomms.2016.10
7. Ahmed, A., Siraj, M.Md., Anazida, Z.: Feature selection using information gain for improved structural-based alert correlation. Plos One **11**(11) (2016). https://doi.org/10.1371/journal.pone.0166017
8. Sun, A., Lim, E., Liu, Y.: On strategies for imbalanced text classification using SVM: a comparative study. Decis. Support Syst. **48**(1), 191–201 (2010). https://doi.org/10.1016/j.dss.2009.07.011
9. Arevian: Recurrent neural networks for robust real-world text classification. In: Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence (2007). https://doi.org/10.1109/wi.2007.126
10. Yoon, K.: Convolutional neural networks for sentence classification. Eprint arXiv (2014). https://doi.org/10.3115/v1/d14-1181
11. Armand, J., Edouard, G., Piotr, B., et al.: Bag of tricks for efficient text classification (2016). https://doi.org/10.18653/v1/e17-2068
12. Zhang, H., Xiao, L., Wang, Y., et al.: A generalized recurrent neural architecture for text classification with multi-task learning. In: Proceedings of the International Joint Conference on Artificial Intelligence (2017). https://doi.org/10.24963/ijcai.2017/473
13. Jagannatha, A., Yu, H.: Bidirectional RNN for medical event detection in electronic health records. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (2016). https://doi.org/10.18653/v1/n16-1056
14. Rahul, D., Salemt, F.M.: Gate-variants of Gated Recurrent Unit (GRU) neural networks. In: Proceedings of the IEEE International Midwest Symposium on Circuits and Systems (2017). https://doi.org/10.1109/mwscas.2017.8053243
15. Chen, J., Li, D., Mirella, L.: Long Short-Term Memory-Networks for machine reading (2016). https://doi.org/10.18653/v1/d16-1053
16. Lai, S., Xu, L., Liu, K., Zhao, J.: Recurrent convolutional neural networks for text classification. In: Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI) (2015)
17. Huang, C., Zhao, H.: Chinese Word segmentation: a decade review. J. Chin. Inf. Process. **21**(3), 8–19 (2007). https://doi.org/10.3969/j.issn.1003-0077.2007.03.002

18. CAIL2018: A large-scale legal dataset for judgment prediction. arXiv preprint arXiv:1807.02478 (2018)
19. Jake, L., Martin, K., Naomi, A.: Points of significance: classification evaluation. Nat. Methods **13**(8), 603–604 (2016). https://doi.org/10.1038/nmeth.3945
20. Yang, Y.: An evaluation of statistical approaches to MEDLINE indexing. In: Proceedings of the Conference of the American Medical Informatics Association (1996). https://doi.org/10.1023/a:1009982220290