Graph Representation Learning for Assisting Administrative Penalty Decisions

Xue Chen[1], Chaochao Liu[2], Shan Gao[3], Pengfei Jiao[4], Lei Du[5], and Ning Yuan[5](✉)

[1] School of Law, Tianjin University, Tianjin 300372, China
[2] Chinese Academy of Cyberspace Studies, Beijing 100048, China
[3] School of Information Engineering, Tianjin University of Commerce, Tianjin 300134, China
[4] School of Cyberspace, Hangzhou Dianzi University, Hangzhou 310018, China
[5] Tianjin Zhongtian Huitong Technology Co., Ltd, Tianjin 300121, China
yuanning@witapex.cn

Abstract. The application of artificial intelligence opens up a new path for administrative punishment and law enforcement, which is of great significance to the modernization of the country's governance capacity. Graph representation learning has been widely used in many judicial scenarios. Most existing administrative legal documents for cause determination and penalty decision are made by means of natural language processing. Due to the many representation methods of information elements in the administrative law enforcement documents, the identification of the cause of action and the decision of punishment are difficult to make, which makes the low accuracy and lack of interpretable. In order to solve these problems, this paper constructs a knowledge graph-based information embedding method to effectively embed knowledge graphs into the network, and builds two graph convolutional neural network frameworks based on node classification and graph classification to realise intelligent assisted case determination and penalty decision based on graph representation. The experimental results show that the graph neural network-based framework is a better choice and the results of multi-task classification are significantly better than using only a single task.

Keywords: Administrative penalties · Graph neural networks · Network embedding · Complex networks

1 Introduction

In July 2017, China's President Xi Jinping gave important instructions on the reform of the judicial system, stressing that "we should follow the laws of justice and combine the deepening of the reform of the judicial system with the application of modern technology". This requires us to deepen the application of modern technology in the judicial field, increase the construction and application of information technology, promote the

Supported by the National Key R&D Program of Jiangxi, China (20212ABC03W12) and National Key R&D Program of China (2020YFC0833303).

Y. Chenggang et al. (Eds.): MobiMedia 2022, LNICST 451, pp. 316–325, 2022.
https://doi.org/10.1007/978-3-031-23902-1_24

upgrading of information technology and intelligence in judicial work, use technology to enhance judicial efficiency, and promote the overall quality and efficiency of trial and execution work.

Deep learning, one of the hottest research directions in the field of machine learning and indeed in computing and the Internet today, has in many ways surpassed previous machine learning algorithms in terms of predictive accuracy since its introduction, showing amazing advantages in image processing, natural language, audio recognition and more. In particular, in recent years, the emergence of convolutional neural networks (CNN) [1], recurrent neural networks (RNN) [2] and adversarial neural networks (GAN) [3] have made deep learning algorithms soar and have received a lot of attention from academia and industry. The emergence of CNN, RNN and GAN in recent years has led to a surge in deep learning algorithms, which have received widespread attention from both academia and industry. The application of deep learning opens up new paths for administrative punishment and law enforcement, and is of great significance to the modernization of the country's governance capacity [4, 5].

In this paper, we construct a knowledge graph-based information embedding method to effectively embed knowledge graphs into the network, and build two graph convolutional neural network frameworks based on node classification and graph classification to realize intelligent assisted case determination and penalty decision based on graph representation. In summary, the contributions of this paper are as follows:

i. We are the first to use graph learning techniques to solve cause inference and sentence penalties.
ii. We propose two frameworks for intelligent assisted penalty decisions based on node classification and graph classification, which can transform administrative legal documents into attribute graphs for adjudication decisions.
iii. Experimental results demonstrate the effectiveness of our proposed framework.

2 Related Work

Current deep learning techniques are becoming increasingly sophisticated, and their end-to-end learning has achieved significant success in text processing tasks by avoiding the error propagation problems associated with heavy feature engineering and natural language processing tools, achieving performance well beyond that of traditional methods [6, 7]. In terms of text feature representation, Mikolov et al. proposed word2vec, a method for training word vectors through neural networks [8]; Later, Joulin et al. proposed fastText [9], an efficient method for text classification and representation learning based on word vectors, using an n-gram model to more effectively represent the relationship between words before and after; The proposed pre-training model of BERT [10] pushed the text feature representation to the top. In text classification matching, Kim proposed the TextCNN approach [11] to apply convolutional neural networks to text classification tasks, which capture key n-gram-like information in sentences through one-dimensional convolutional kernels; Liu et al. proposed a network design that uses RNNs for classification problems [12], taking into account the temporal characteristics of text; A number of network variants, such as LSTM, RCNN, and network models that introduce the attention [13] mechanism, have since emerged.

In recent years, with the continuous disclosure of judicial data represented by judicial documents and the breakthrough of natural language processing technology, how to apply artificial intelligence technology in the judicial field to improve the efficiency of judicial personnel in the case processing process has gradually become a hot topic of legal technology research, and some scholars have been studying the legal text processing technology related to deep learning. Luo et al. [14] proposed a neural network approach based on an attention mechanism to integrate the law information into the charge prediction task to rationalize the charge prediction more and help improve the efficiency of the legal assistant system. Hu et al. [15] proposed an attribute-attentive offense prediction model to classify offenses according to legal attributes and manually tag relevant offense attribute information to significantly improve the prediction accuracy of low-frequency and confusing offenses. Zhong et al. [16] used directed acyclic topological graphs to model logical dependencies among multiple tasks, fusing statutes, charges and sentences into a unified judicial sentencing framework, achieving consistent and significant improvements in the effects of all tasks across multiple datasets. To our knowledge, we have not seen a relevant area where graph neural networks have been used to intelligently assist in penalty decisions.

3 The Proposed Framework

This paper proposes two frameworks for assisting penalty decisions from node classification and graph classification perspectives.

3.1 The Framework for Intelligent Assisted Penalty Decisions Based on Node Classification

This framework constructs the entire administrative penalty instrument data as an administrative penalty network, where each administrative penalty instance is a node and the connected edges between nodes are determined by the relevance of the type of offence and the fact of the offence between penalty instances. Specifically, for single task classification, the penalty category of each administrative penalty instance is used as the label of the node to train the node classification model; for multi-task classification, the penalty category and penalty basis of each administrative penalty instance are used as different labels of the node to train the multi-task model; for multi-label fusion, the penalty category and fine amount of each administrative penalty instance are fused into one label.

3.1.1 Label Generation

The label generation rules are as follows:

(1) The generation of penalty category label: In the “penalty category” field, each category is divided by a semicolon. Firstly, all the penalty categories are counted and labelled numerically. Secondly, one hot vector is generated for each administrative instrument, setting the position of the penalty category to which the administrative instrument is set to 1.

(2) Fine amount label generation: As each administrative instrument penalty instance involves a different fine amount, the fine amount interval is set for labeling. The interval division rules are shown in Table 1 and Table 2.

Table 1. Label division rules of fine amount (unbalanced label)

Amount of penalty ($ million)	Label
0	0
0–1	1
1–10	2
10–50	3
50–100	4

Table 2. Label division rules of fine amount (balanced label)

Amount of penalty ($ million)	Label
0	0 (20000 records)
0–0.1	1 (20972 records)
0.1–0.5	2 (14596 records)
0.5–10	3 (12020 records)
>10	4 (10000 records)

(3) Penalty basis label generation: In the "penalty basis" field, there are specific legal provisions. Firstly, the legal basis of all instruments are counted and labeled numerically. Secondly, one-hot vectors are generated for the legal basis of each administrative instrument, and the corresponding position of it involved in the administrative instrument is set to 1.

3.1.2 The Generation of Sentence Vectors

Two different approaches are used for vector generation at this stage. The first method is generated by word2vec; the second is generated by Bert. Firstly, using python's jieba splitting, the type of offence and the fact of offence for each instance of the administrative instrument are split; secondly, the splitting results are eliminated by stop words; finally, the results are input into the two models respectively to obtain the sentence vectors of the corresponding models.

3.1.3 Graph Structuring Stage

The cosine similarity between any two sentence vectors is first calculated from the obtained sentence vectors, and a graph adjacency matrix with weights is constructed from the similarity matrix. Where each case is treated as a node in the graph. The cosine similarity can be formulated as follows:

$$\cos(\theta) = \frac{\sum_{i=1}^{n}(x_i \times y_i)}{\sqrt{\sum_{i=1}^{n}(x_i)^2} \times \sqrt{\sum_{i=1}^{n}(y_i)^2}} \tag{1}$$

Next, the interval corresponding to the weights is set and the resulting cosine similarity matrix is converted into an adjacency matrix with a weighted graph. The weight correspondences are shown in Table 3.

Table 3. Weight corresponding conversion table

Cosine similarity interval	Weight
0–0.5	0
0.5–0.6	1
0.6–0.7	2
0.7–0.8	3
0.8–0.9	4
0.9–1	5
1	0

3.1.4 The Stage of Graph Neural Network

We built a two-layer GCN [19] model to learn the administrative penalty document node vectors, which are then classified by a one-layer MLP. For multi-label classification, the penalty category and penalty are classified by two different MLP according to two different labels.

A two-layer *graph neural network is as follows:*

$$Z = f(X, A) = softmax(\hat{A}ReLU(\hat{A}XW^{(0)})W^{(1)}) \tag{2}$$

where $\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$, X is the attribute matrix.

3.2 The Framework for Intelligent Assisted Penalty Decisions Based on Graph Classification

Each administrative penalty instance is constructed as a network, and the nodes of the network are phrases in the text of the type of offence and the fact of the offence in that penalty instance. The connected edges between the nodes are determined by the correlation between the phrases, and the word vectors of the phrases are used as attributes of the corresponding nodes. The penalty category of each administrative penalty instance is used as the label of the corresponding network to train the graph classification model. The final result is a graph mining-based intelligent aid for cause determination and penalty decision.

Specifically, the label generation rules and word and sentence vector generation phases are the same framework as the intelligent assisted penalty decision based on node classification, differing mainly in the graph generation and word vectorisation, and graph neural network phases.

3.2.1 Graph Generation and Word Vectorisation

Each case is constructed as a word network, with each word being a node, by calculating the PMI values between words in each case (text). The word2vec model is used to find the representation vector of each word in the word network, which is used as the attribute vector of that word. The PMI is given by the following formula.

$$PMI = log\frac{p(i, j)}{p(i)p(j)} \tag{3}$$

where $p(i, j) = \frac{\#W(i,j)}{\#W}$, $p(i) = \frac{\#W(i)}{\#W}$. $\#W(i)$ refers to the number of sliding windows which contain the word i, $\#W(i, j)$ is the number of sliding windows containing both the word i and the word j, $\#W$ refers to the total number of sliders in the corpus.

3.2.2 The Stage of Graph Neural Network

For graph classification, we learn graph vectors through a two-layer GCN [19] model and a readout layer [18], and then classify them through a layer of MLP.

4 Results

In this section, we use administrative penalty instrument data to verify the effectiveness of our proposed framework. The experimental results show that the graph neural network-based framework is a better choice.

4.1 Evaluation Metric

We use Accuracy [20] and Hamming [21] to evaluate our framework performance. Two metrics are defined as follows:

(1) The accuracy is defined as the proportion of the total sample that is correctly predicted, namely

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{4}$$

(2) Hamming is used to measure the number of times a label is misclassified, i.e. a label belonging to a sample is not predicted and a label that does not belong to that sample is predicted to belong to that sample.

$$\mathrm{Hamming_{loss}} = \frac{1}{n}\sum\nolimits_{i=1}^{n} \frac{1}{m}\left|h(x_i)\Delta y_i\right| \tag{5}$$

the single-task node classification task uses two different metrics: $\mathrm{Hamming_{loss}}$ and Accuracy. We use 10% of task metrics for $\mathrm{Hamming_{loss}}$ and the other task metrics are for Accuracy. We use 10% of datasets for the validation and 20% for the test set. Obviously, the higher accuracy means the better performance, while the smaller the loss, the better for Hamming loss.

4.2 Parameters Setting

For the sentence vector generation phase, we follow the base settings of both papers, where word2vec [17] has a sentence vector dimension of 300 and Bert [10] has the dimension of 768. For the node classification and graph classification tasks, we set the number of iterations of the GCN [19] model to 200 and 100, respectively, and the output node vector and graph vector dimension set to 128.

4.3 Datasets

The administrative penalty instrument dataset is used in this dataset. 82,363 records are recorded in this dataset, and 71,457 data were obtained after data cleaning for single-task node classification, label fusion node classification and graph classification, and 65,374 data were used for multi-task node classification. The number of processed dataset is 71,457.

4.4 Experiment Analysis

In this section, we explore the performance of the model in three directions: single-label, multi-label and multi-label fusion.

Figure 1 gives the results of sentencing penalties for single-label administrative instruments based on node classification and graph classification. (a) and (b) correspond to the results of the node classification. The horizontal axis represents the proportion of the training set and the vertical axis corresponds to the evaluation index. As can be seen from Fig. 1, the performance of the model tends to increase as the training ratio increases. Good results were achieved for both the node classification and graph classification tasks, indicating that the graph neural network based approach is a better choice.

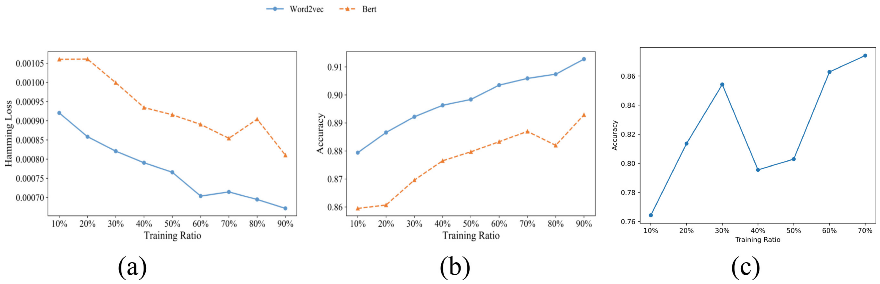


Fig. 1. Results of sentencing penalties for single-label administrative instruments

Figure 2 gives the results of multi-labelled administrative instruments based on node classification for sentencing penalties. It is obvious to see from the figure that the results for multi-task classification are significantly better than using only a single label.

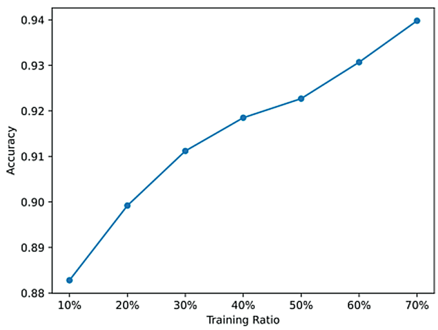
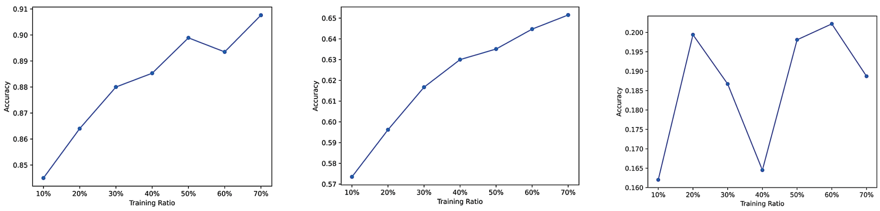


Fig. 2. Results of sentencing penalties in multi-label administrative instruments

Figure 3 shows the results of sentencing penalties in multi-label fusion administrative instruments. The three graphs correspond from left to right to label fusion node classification (unbalanced), label fusion node classification (balanced), and label fusion graph classification. When fusing the penalty category and the fine amount of the sample into one label, similar results to the single task are achieved, but when equalizing the division of the fine amount labels, the results are unsatisfactory. The equalization of the penalty amount labels caused an imbalance in the fused label categories and so led to poorer final results.

Overall, our model achieved good results for both the node classification and graph classification tasks, illustrating that a graph neural network-based approach is a better choice.



(a) (b) (c)

Fig. 3. Results of sentencing penalties in multi-label fusion administrative instruments

5 Conclusion

In this paper, we propose two intelligent assisted punishment frameworks, which are graph neural network frameworks based on node classification and graph classification respectively. To the best of our knowledge, this is the first time that graph learning techniques have been used to solve such problems. Overall, good results are achieved for both the node classification and graph classification tasks, indicating that the graph neural network approach is a better choice. The results for multi-task classification are significantly better than using only a single label. Similar results to the single task are achieved when fusing the penalty category and the fine amount of the sample into one label, but poorer results are achieved when the fine amount of labels are equalized. The equalization of the penalty amount labels caused an imbalance in the fused label categories and so led to poorer final results. A possible improvement solution would be to process the fused labels by removing some of the samples from a few categories or by combining them into one category.

References

1. Alzubaidi, L., Zhang, J., Humaidi, A.J., et al.: Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J. Big Data **8**(1), 1–74 (2021)
2. Nasir, J.A., Khan, O.S., Varlamis, I.: Fake news detection: a hybrid CNN-RNN based deep learning approach. Int. J. Inf. Manag. Data Insights **1**(1), 100007 (2021)
3. Zhou, F., Yang, S., Fujita, H., et al.: Deep learning fault diagnosis method based on global optimization GAN for unbalanced data. Knowl. Based Syst. **187**, 104837 (2020)
4. Raaijmakers, S.: Artificial intelligence for law enforcement: challenges and opportunities. IEEE Secur. Priv. **17**(5), 74–77 (2019)
5. Berk, R.A.: Predictive policing, and risk assessment for law enforcement. Annu. Rev. Criminol **4**, 209–236 (2021)
6. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
7. Young, T., Hazarika, D., Poria, S., et al.: Recent trends in deep learning based natural language processing. IEEE Comput. Intell. Mag. **13**(3), 55–75 (2018)
8. Mikolov, T., Sutskever, I., Chen, K., et al.: Distributed representations of words and phrases and their compositionality. In: Proceedings of the 26th International Conference on Neural Information Processing Systems, pp. 3111–3119 Lake Tahoe, NV, USA (2013)
9. Joulin, A., Grave, E., Bojanowski, P., et al.: Bag of tricks for efficient text classification. In: Proceedings of the 15th Conference of the European Chapter of The Association for Computational Linguistics, pp. 427–431, Valencia, Spain (2017)

10. Devlin, J., Chang, M.W., Lee, K., et al.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of 2019 Conference of the North American Chapter of the Association for Computational Linguistics, 4171–4186, Minneapolis, MN, USA (2019)
11. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of 2014 Conference on Empirical Methods in Natural Language Processing, pp. 1746–1751, Doha, Qatar (2014)
12. Liu, P.F., Qiu, X.P., Huang, X.J., et al.: Recurrent neural network for text classification with multi-task learning. In: Proceedings of the 25th International Joint Conference on Artificial Intelligence, 2873–2879, New York, NY, USA (2016)
13. Vaswani, A., Shazeer, N., Parmar, N., et al.: Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 5998–6008, Red Hook, NY, USA (2017)
14. Luo, B.F., Feng, Y.S., Xu, J.B., et al.: Learning to predict charges for criminal cases with legal basis. In: Proceedings of 2017 Conference on Empirical Methods in Natural Language Processing, pp. 2727–2736, Copenhagen, Denmark (2017)
15. Hu, Z.K., Li, X., Tu, C.C., et al.: Few-shot charge prediction with discriminative legal attributes. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 487–498, Santa Fe, NM, USA (2018)
16. Zhong, H.X., Guo, Z.P., Tu, C.C., et al.: Legal judgment prediction via topological learning. In: Proceedings of 2018 Conference on Empirical Methods in Natural Language Processing, pp. 3540–3549, Brussels, Belgium (2018)
17. Goldberg, Y., Levy, O.: word2vec explained: deriving Mikolov et al.'s negative-sampling word-embedding method. arXiv preprint. arXiv:1402.3722 (2014)
18. Yao, L., Mao, C., Luo, Y.: Graph convolutional networks for textclassification. In: Proceedings of the AAAI conference on artificial intelligence, vol. 33, no. 1, pp. 7370–7377 (2019)
19. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint. arXiv:1609.02907 (2016)
20. Kaya, İ, Kahraman, C.: Development of fuzzy process accuracy index for decision making problems. Inf. Sci. **180**(6), 861–872 (2010)
21. Butucea, C., Ndaoud, M., Stepanova, N.A., et al.: Variable selection with Hamming loss. Ann. Stat. **46**(5), 1837–1875 (2018)