International Journal for Housing Science and Its Applications
Publish August 10, 2025. Volume 46, Issue 4 Pages 5030-5039

https://doi.org/10.70517/ijhsa464346

Entity Relationship Extraction from Legal Texts Based on Graph Neural Networks

Amin Wang[1,*]
[1] Institute of Marxism, Zhengzhou Tourism College, Zhengzhou, Henan, 451464, China
Corresponding authors: (e-mail: ldlj2024@126.com).

Abstract Legal texts usually contain complex entity relationships, and traditional manual analysis methods are not only inefficient but also easily affected by human factors. In this study, a new entity relationship extraction model for legal texts based on graph convolutional networks and BERT, named ON-BERT, is proposed. The model captures hierarchical semantic features in the text through the hierarchical structure parsing module and extracts global semantic information by combining with the BERT pre-trained language model. The experiments are conducted on 15,000 criminal judgments published in China Judgment Website, and 12,163 valid case texts are obtained after data processing. The experimental results show that the ON-BERT model outperforms the traditional model in terms of precision, recall and F1 value. In the test, the F1 value of ON-BERT is 83.56%, which is improved by 3.92% compared to the BERT model, and in terms of accuracy, ON-BERT also significantly outperforms the other models, reaching 82.55%. In addition, ON-BERT also shows significant improvement in training efficiency and inference speed, and its training time is shortened by about 4 times compared to the baseline model. The effectiveness and efficiency of this model provides a new technical path for legal text analysis.

Index Terms Legal Text, Entity Relationship Extraction, Graph Convolutional Networks, BERT, Hierarchical Parsing, ON-BERT

I. Introduction

With the profound changes in the judicial field, legal documents have become more and more important in the judicial informatization project [1]. A large number of legal documents have not only become a favorable guarantee for Chinese courts to promote the disclosure of judicial information and the fairness and justice of justice, but also provide a data basis for the judicial reform and the construction of “smart courts” in various places [2]-[4]. In the face of the huge amount of legal documents, how to use artificial intelligence technology to help lawyers and related researchers better summarize the laws of the law has become an important issue of data mining in the judicial field.

As a kind of textual data, legal text can not be done to be recognized directly by computer, so it is often necessary to transform the text into formatting [5], [6]. Entity-relationship extraction technology can organize the discrete entities in the text into structured data in the format of entity-relationship-entity triad, which intuitively reflects the relationship between entities [7]. As a result, entity-relationship extraction technology is applied to the legal field, which can quickly extract the key elements from various legal text data, so that the decision makers can intuitively feel the content of the law-related information, focusing on the data mining of the relevant information in the legal text [8]-[11]. At the same time, with the help of visualized relational network to show the relationship between entities and entities, it provides an important basis for the government to carry out rights enforcement, case monitoring and decision-making [12], [13].

The ON-BERT model proposed in this paper makes up for the shortcomings of existing models in multi-level information extraction by combining hierarchical structure parsing and global semantic representation of BERT. This model first learns to construct the graph structure of legal texts through graph representation and learns the hierarchical features in sentences by combining the hierarchy parsing module. Next, the global semantic information of the text is extracted using the BERT model, and the semantic links between nodes in the text are further enhanced by graph convolutional neural network. Finally, the information at different levels is weighted by the hierarchical attention mechanism, so as to effectively capture the relationships between different entities in the sentence. In order to verify the effectiveness of the method, the study designed several sets of comparative experiments to evaluate the performance of the ON-BERT model in legal text entity relationship extraction by comparing it with the existing ON-LSTM model and BERT model.

II. Relationship extraction modeling

In this chapter, a new relationship extraction model ON-BERT is proposed, and a hierarchical structure parsing module is designed based on the idea of ordered neuron ON-LSTM, which learns the hierarchical structure of the input sentence so as to capture the semantic features at different levels in the sentence, and at the same time, a new attention mechanism - hierarchical attention is proposed, which helps to pay attention to the information at different levels more effectively so as to more comprehensively understand the structure of the sentence, and the learned hierarchical structure information will guide the model to accomplish the legal text entity relationship extraction task.

II. A. Graph representation learning and GCN networks

II. A. 1) Graph representation learning

The main purpose of graph representation learning is to transform graph data into low-dimensional, dense vector representations and to ensure that the properties of the graph data are maintained in the vector space. Currently, there are three main types of graph representation learning methods, namely, matrix decomposition-based graph representation learning methods, random walk-based graph representation learning methods, and deep learning-based graph representation learning methods [14].

Matrix decomposition based methods are based on the decomposition of the network adjacency matrix, such as SVD, PCA, NMF, etc. These methods can be used for dimensionality reduction and denoising, but usually cannot capture complex network topologies. Methods based on random wandering are DeepWalk algorithm, Node2vec algorithm, etc. These methods obtain the contextual information of nodes by performing random walks on the graph structure and use this information to learn the embedding representation of nodes, which can preserve the local structural information of the network and have excellent performance in dealing with sparse networks, but these methods are difficult to deal with directed graphs, the quality of the node embedding is greatly affected by its initialization, and it is very difficult to adjust the parameters of the network. Deep learning based graph representation learning methods automatically learn node embedding representation through the model, without the need to manually design feature engineering, and deep learning methods are very flexible, can adapt to a variety of graph structure data and tasks, can handle a large amount of data, and also has a better ability to deal with sparse graph data, and at the same time, the deep learning methods can increase the number of layers to improve the performance of the model, and can be expanded. Strong performance. The graph representation learning method based on deep learning to explore the entity relationship of legal text can show a strong modeling ability and characterization ability.

II. A. 2) GCN network

Graph Convolutional Neural Network (GCN) is a neural network model based on the idea of "convolution" applied to graph data. It views the graph data as a set of nodes and edges connecting the nodes, and uses the convolutional approach to transfer information and feature extraction on the graph structure to learn the node representation [15]. The feature vector of each node in the GCN is obtained from the weighted summation of the feature vectors of its neighboring nodes, and then added to its own feature vector, and the computation process is shown below:

$$H^{(l+1)} = \sigma(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(l)}w^{(l)}) \tag{1}$$

$$\hat{A} = A + I \tag{2}$$

where H^{l} refers to the input features of the l th layer and H^{l+1} refers to the output features. w^{l} is the linear transformation matrix. $\sigma(\cdot)$ is the nonlinear activation function. $\hat{A}$ is called the adjacency matrix with self-connections, referred to as the self-connecting adjacency matrix, and is usually denoted by the adjacency matrix A, which is defined as shown in public notice (3), where I is the unit matrix. The A_{ij} in the adjacency matrix is whether or not there is a path between node i and node j, with 1 being yes and 0 being no. The $\hat{D}$ is the degree matrix of the self-connecting matrix, which is defined as:

$$\hat{D}_{ij} = \sum_{j}\hat{A}_{ij} \tag{3}$$

where $\hat{D}^{-\frac{1}{2}}$ is the square root taking the inverse of the self-connectivity matrix.

Thus, in Eq. (2) of GCN, $\hat{o}^{-\frac{1}{2}\tilde{A}D-\frac{1}{2}\tilde{A}D}$ these are actually computed from the adjacency matrix A, so you can think of these as a constant. All the model needs to learn is the weight matrix w^l. After understanding this formula, all that is needed is to construct a graph, count the adjacency matrices, and substitute them directly into the formula to implement a GCN network.

II. B.BERT pre-trained language model

Research on obtaining linguistic representations based on pre-training has made great strides over time. Neural network-based word embedding learning has also been applied very successfully. These studies provide support for high-quality initialized word vectors and improve the efficiency and speed of model training. By using the language model as a training task and performing unsupervised training on a huge amount of unlabeled text, the parameters of the model can be learned and further fine-tuned in the task-specific model to improve the performance of the model to a greater extent.

The input layer in the BERT model contains three parts, which are static word vector coding, positional coding and utterance segmentation coding, that is, each Token contains word information, positional information and information of the passage in which it is located. These three kinds of embedding are obtained by learning.

BERT adopts Transformer's encoder structure as a feature extractor and uses the accompanying masked language model MLM training method, this pre-training method is to randomly mask a number of words in the input sequence and require the model to predict the word vectors of these masked words at the time of encoding, thus forcing the model to simultaneously take into account all the words in the input sequence, and realizing the bi-directional text of input sequence encoding [16].

The Transformer encoder is a module consisting of a multilayer self-attention mechanism and a feedforward neural network. The self-attention mechanism enables the model to simultaneously consider all words in the input sequence and generate a context-sensitive word vector representation for each word. Therefore, with the multi-layer self-attention mechanism, BERT can capture richer and more complex contextual information. In addition, the pre-training task of the masked language model also enables BERT to mask some words randomly in the input sequence and ask the model to predict the word vector representations of these words, which further improves the model's learning ability for contextual information.

The main advantage of BERT over other neural network-based natural language processing models is its use of large-scale unlabeled data for pre-training, which allows for fine-tuning and optimal performance on a variety of natural language processing tasks. In addition, BERT's bidirectional encoder structure and multi-layer self-attention mechanism enable it to simultaneously consider contextual information in the input sequence, resulting in a stronger ability to extract semantic information.

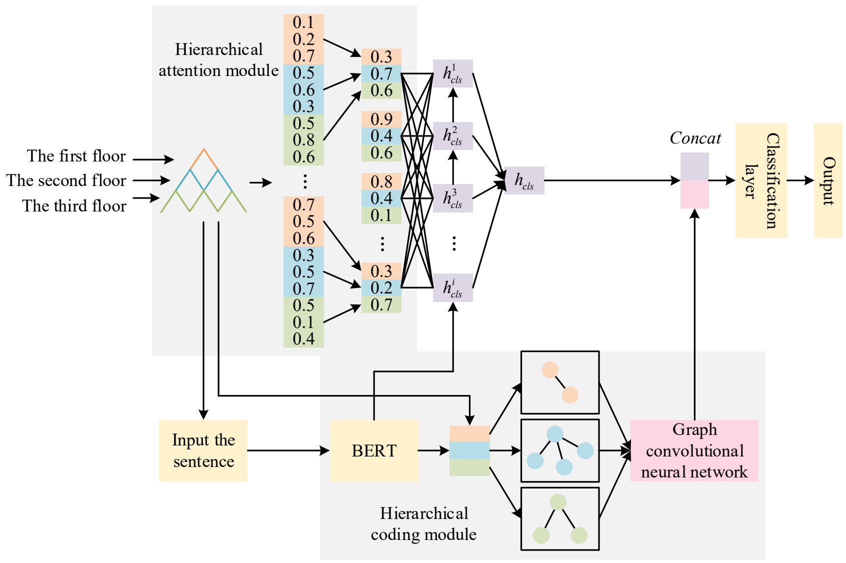


Figure 1: ON-BERT model diagram

II. C.Modeling

II. C. 1) Overall model architecture

ON-BERT performs hierarchical modeling of the input sentence through the hierarchical representation module and obtains semantic features at different levels. The model is based on the global semantic information extracted by BERT, which is fused with the hierarchical structure information learned by the hierarchical structure parsing module through the attention weights generated by dot product to form a feature vector containing the overall hierarchical structure of the sentence and the global semantic information. Meanwhile, the graph representations of different levels are integrated with the entity state vectors generated by BERT and encoded by graph convolutional neural network to obtain another state vector. Finally, these two are spliced and classified by Softmax to complete the relationship extraction task.The ON-BERT model is shown in Fig. 1. The model consists of the following two main modules: hierarchical attention layer and hierarchical coding layer.

II. C. 2) Hierarchical Attention Layer

(1) Hierarchy parsing module

This chapter designs a hierarchy parsing module based on the idea of ordered neuron ON-LSTM, which enables the model to learn hierarchical information during the training process. When a sentence is input, it is decomposed into a sequence of words. Each word is mapped into a high-dimensional vector through an embedding layer, which captures the semantic information of the word. These embedding vectors are then fed into input gates f_t, forgetting gates i_t, and output gates o_t to regulate the flow of information, where the three gate structures correspond to Eqs. (4) ~ (6), respectively:

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

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{5}$$

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

where f_t, i_t and o_t denote the forgetting gate, the input gate, and the output gate at step t, respectively, x_t denotes the inputs at step t, and h_{t-1} denotes the hidden state of the previous step. In Eq. (7) the pass gate x_t will be integrated into $\hat{c}_t$ defined as the memory state and weighted and summed with the previous three gates, and the formula for the above step is as follows:

$$\hat{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{7}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \hat{c}_t \tag{8}$$

$$h_t = o_t \circ \tanh(c_t) \tag{9}$$

The parsing module divides the input word sequence into different levels, each with a corresponding LSTM unit, and a tree structure is formed between these units. Each LSTM unit receives the hidden state of the previous level as input, and captures different semantic information at different levels by learning to remember which information is important for modeling the current level. Previous information can be selectively forgotten through forgetting gates, while input gates control the input of new information. This allows the model to capture long-distance dependencies in a sentence, leading to a better understanding of contextual relationships. These gates are computed and integrated into the LSTM unit as shown below:

$$\hat{f}_t = cummax(w_{\hat{f}} x_t + U_{\hat{f}} h_{t-1} + b_{\hat{f}}) \tag{10}$$

$$\hat{i}_t = 1 - cummax(w_i x_t + U_i h_{t-1} + b_i) \tag{11}$$

For $cummax$ in Eq. (10), which is an activation function defined as $cummax(x) = cumsum(softmax(x))$, $\hat{f}_t$ in Eq. (12) and $\hat{i}_t$ in Eq. (13) denote the forgetting gate and input gate in the representation layer differs from the forgetting gates f_t and input gates i_t in the representation layer because gates in the ordinary LSTM assume that the neurons in their hidden vectors are equally important and that these neurons are active for every word in the sentence:

$$\overline{f_t} = \hat{f}_t \circ (f_t \hat{i}_t + 1 - \hat{i}_t) \tag{12}$$

$$\overline{i_t} = \hat{i}_t \circ (i_t \hat{f}_t + 1 - \hat{f}_t) \tag{13}$$

$$c_t = \overline{f_t} \circ c_{t-1} + \overline{i_t} \circ \hat{c}_t \tag{14}$$

(2) Hierarchical Attention Generation Module

This section proposes a novel attention formula that generates an attention score matrix by computing the dot product between the hierarchical information matrix and the representation vector of the CLS sequence of the BERT intermediate layer when fusing the hierarchical information and the CLS sequence of the BERT intermediate layer. This matrix reflects the level of attention paid to each position in the BERT sequence at each time step of the hierarchy parsing module. A normalized attention weight matrix is obtained by performing a Softmax operation on the attention score matrix.

Finally, this weight matrix was applied to the hierarchical information matrix and the fused representation was obtained by weighted fusion. This process allows the model to dynamically adjust its attention to different parts of the hierarchical information and the BERT while processing textual tasks, thus better utilizing the information between the two and improving the model's representational capabilities. The process of obtaining the attention matrix by dot-producting the hierarchical information and the CLS sequence of the BERT middle layer is as follows:

$$a_{ti} = \frac{\exp(W_h h_{cls}^{t-1} + b_h)\cdot(W_x x_i + b_x)}{\sum_{j=1}^{N}\exp((W_h h_{cls}^{t-1} + b_h)(W_x x_j + b_x))} \tag{15}$$

where W_h, b_h, W_x and b_x are learnable parameters. In this formulation, the attention weight of each word is computed using the hidden vector h_{t-1} from the parsing module in the previous step as the query vector. The basic principle is to enrich the attentional weights of the current step with contextual information from previous steps, thus producing a contextual input representation x_t with richer feature information and importance scores computed in the hierarchical parsing module:

$$H_s = h_{cls} a_{ti}^T \tag{16}$$

$$H_s^{'} = W_e(\tanh(H_s)) + b_e \tag{17}$$

Ultimately, the fused representation $H_s^{'}$ obtained by the above computational formula (17) can be used for subsequent relationship extraction tasks. Through dot product and Softmax operations, the dynamic adjustment of the attention levels of hierarchical information and BERT intermediate layer CLS sequences is realized, which enables the establishment of weighted correlations between the two, and better utilizes the information of the two models. This dot product approach to obtaining the attention matrix allows the model to dynamically adjust the degree of attention to different parts of the input text according to the contextual information of the input text, improving the model's representational capabilities.

II. C. 3) Hierarchical coding layers

The last hidden state of BERT is used in the hierarchical coding layer to encode the graph via GCN to obtain the final vector $H_k^{'}$, for the A^k represented by the kth layer of the hierarchy, where the i th node feature in the first layer is represented as:

$$h_k^l = \sigma\left(\sum_{j=1}^{n} A_{ij}^k W_k^l h_i^{l-1} + b_k^l\right) \tag{18}$$

where W_k^l and b_k^l are the weight matrix and bias vector of the corresponding hierarchy in the kth layer of layer 1, respectively. σ is the activation function Relu.

II. C. 4) Relationship classification layer

At the relationship classification layer, the vectors from the attention layer of the hierarchy and the hierarchical coding layer are first spliced. This spliced vector contains contextual information from the attention mechanism and global semantic information from the hierarchical coding. Then, a linear transformation is performed through this spliced vector and the scores are converted into a probability distribution by means of a softmax activation function. This probability distribution represents the predicted probability of each relationship category. Eventually, the category with the highest probability is selected as the result of relationship categorization in order to complete the relationship extraction task. The specific formula is as follows:

$$h_{final} = W_o[concat(H_{s'}, H_{k'})] + b_o \tag{19}$$

$$p(r \mid X;\theta) = softmax(h_{final}) \tag{20}$$

W_o and b_o denote the weight matrix and bias vector, respectively, and h_{final} is the final output vector representation. This process enables the model to synthesize the local and global information of the text and capture the semantic relationships between entities more effectively.

III. Analysis of the substantive relationship of legal texts

III. A. Data sets

The case text in legal documents reflects the process and decision results of the court, and in the current research on entity relationship extraction, there is no standard dataset on entity relationship extraction of case text. At present, there are more than one million public cases on the Internet, and the relevant case data are easier to obtain. In order to ensure the authenticity and reliability of the experiment, this paper selects some of the criminal documents published by China Judicial Instruments Network (CJIN) during the period of 2021-2024, the content of which mainly consists of criminal verdicts and criminal judgments, with a total of 15,000 articles, and removes punctuation, special characters, comments, redundant words, and retains only 12,163 case texts of moderate length.

In this paper, the dataset used for training, validation, and testing required for the experiment is built by manual annotation. First, top-level entities (in this paper, only criminal judgments) are added before each sentence and segmented by @ and the original text. When extracting the corpus, if the triples originate from multiple sentences above and below, the sentences are spliced and the Combined field in the “spo_list” records the splicing information. If there are more than one triple in the sentence, it will be recorded in the “spo_list” field, and the source text will be recorded in the “text” field.

III. B. Experimental content

In order to verify the specific performance of hierarchical attention module and hierarchical coding module on the model, the model is experimented on the case labeled dataset established in this paper, and three sets of comparison experiments are set up in this chapter:

(1) ON-LSTM. The hierarchical attention module is removed in the experiments, i.e., the ordered neuron ON-LSTM is replaced with Bi-LSTM for implementation.

(2) BERT. The hierarchical coding module was removed from the experiment, i.e., BERT was replaced with Glove to train word vectors.

(3) ON-BERT. Using the model proposed in this chapter, by learning the hierarchical structure and global information of the text and modeling the hierarchical structure.

III. C. Experimental results and analysis

III. C. 1) Comparative Experimental Analysis

The models in this paper introduce a hierarchical attention module and a hierarchical coding module, and a comparison of the results of the three sets of experiments is shown in Table 1. The iteration time is the time corresponding to when each model gets the highest accuracy in the test set. The ON-BERT model proposed in this paper outperforms the other two models in terms of accuracy, F1 value. Comparing from the table downwards, it can be seen that the iteration time cost of the BERT model is smaller than that of the ON-LSTM model, and the comparison of the two can be seen that the BERT network structure is superior to the ON-LSTM network structure in terms of saving the training time and improving the training efficiency, etc. The BERT is not only computationally simpler but also converges faster, which can improve the accuracy while reducing the model's training time. Comparing the BERT and ON-BERT models, both models use the BERT network structure to extract important feature information in the text, and the difference is that the latter adds one more word-level self-attention layer, which improves the accuracy and F1 value by 3.92% and 4.3%, respectively. Due to the addition of one more attention layer, the weighted computation time is increased while highlighting the important information. The comparison clearly shows the improvement of hierarchical attention on the model's effectiveness.

Table 1: Experimental comparison

Model	Precision	Recall	F1	Time/s
ON-LSTM	76.34%	81.63%	79.26%	1463
BERT	78.63%	82.44%	80.63%	1236
ON-BERT	82.55%	85.98%	83.56%	1322

III. C. 2) Analysis of the number of iterations

In order to better analyze the relationship between the dynamic changes of the models and the number of iterations during the experiment, 50 iterations of the experiment were selected for the analysis. The relationship between the accuracy, F1 value and the number of iterations obtained from the three groups of models on the test set is plotted to show the change of accuracy with the number of iterations as shown in Fig. 2. The horizontal coordinate of the graph is the number of iterations and the vertical coordinate indicates the accuracy value of the experiment. The graph demonstrates the change in the accuracy rate of the model with the increase in the number of iterations, the accuracy rate of the three models are increasing with the increase in the number of iterations, the accuracy rate of the ON-BERT model proposed in this chapter is significantly improved after the 23rd iteration, and the model achieves the optimal value of the accuracy rate in the interval of 45 to 50 iterations, and the accuracy rate is better than that of the other two models in the overall performance.

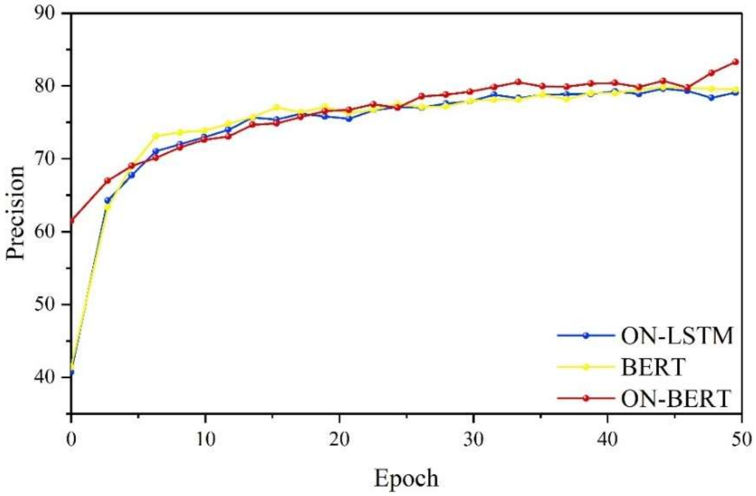


Figure 2: The accuracy value varies with iteration

When the accuracy of ON-BERT and BERT models reaches the optimal value, the accuracy of different kinds of relationships is shown in Table 2. The two models have the highest recognition accuracy for the “Has_Crime” and “Has_Sentence_Outcome” relationships, which are both over 90%, probably because the characteristics of the relationships between the person and the crime type entity are easy to be found. The lowest accuracy was for relationships between persons, which may be due to the small number of relationships in the dataset, which were predicted to be other relationships during the training process. In the dataset, for the same person's name, it may appear several times, the form of the person's name is not uniform, and “so-and-so” will be used instead of the real name, which leads to a lower accuracy rate. After adding the hierarchical coding module, the accuracy of the model for each type of relationship is improved, and the biggest improvement is for the relationship “Has_Crime”, with an increase of 3.23%, which indicates that the model is superior in recognizing the unique criminal charges in the text of the cases in the field of law, and taking into account the feature information of the words, so that the model can recognize professional nouns in the text of the cases. This indicates that the model is superior in recognizing criminal charges in case texts in the legal field, taking into account the feature information of the words, so that the model is more accurate in recognizing the nouns in case texts.

Table 2: Accuracy of physical correlation

Relational type	ON-BERT	BERT
Character relation	46.93%	47.21%
Has_Offence	96.33%	99.56%
Has_Verdict	97.22%	98.96%
Has_According to the rule	71.11%	72.63%
Has_Light plot	84.66%	85.97%

The variation of F1 values with the number of iterations is shown in Fig. 3. The F1 of the models proposed in this chapter during the iterative experiments are all higher than the other two models in the comparison test, with the highest F1 value reaching 83.56%. The F1 value is improved by 2.93% over the benchmark model BERT. The F1 value of each model increases with the number of iterations and finally converges and stabilizes. There are some fluctuations in the model iteration process, but the degree of fluctuation is not significant.

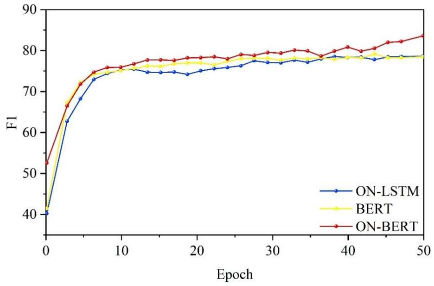


Figure 3 F1 values vary with iteration times

III. C. 3) Efficiency analysis

In practice, the named entity recognition task focuses on accuracy and also needs to focus on efficiency in order to realize further commercial applications. To explore the efficiency of the proposed models in this chapter in practice, further experiments are conducted to explore the training and decoding times of the baseline and ON-BERT models during training and decoding. The experimental results are shown in Table 3. The model in this paper is 4 to 8 times faster than the baseline in terms of training and decoding speeds, respectively, and the ON-BERT model significantly improves the efficiency of the inference process and is sufficiently competitive in terms of efficiency. Through analysis, it is found that the efficiency improvement comes from the convolutional operator, which has the advantage of fast computation speed when dealing with high-density multi-dimensional data.

Table 3: The average time of training and decoding iterations in one epoch on dataset

Model	Train	Dev
BERT	100s	24s
ON-BERT	23s	3s

III. C. 4) Experimental analysis of sentence lengths

In order to investigate the performance ability of the model in long sentences and to prove that the model in this chapter has better performance in complex utterances, this subsection tests different sentence lengths, and the test results are shown in Figure 4. The test set is categorized into three categories (0, 50], (50, 100], (100-) based on sentence length. It can be seen that the accuracy of both the model in this paper and the large language model (Llama2) decreases with the increase of the input sentence length, which is most likely due to the fact that longer sentences correspond to more complex dependency structures, which increases the difficulty of relation extraction, which is in line with the general perception, and it also fully proves that focusing on complex utterances, such as long sentences, is of great relevance for the improvement of the accuracy of the relation extraction task. Meanwhile, it can be clearly found that ON-BERT consistently outperforms Llama2, and the gap is very obvious in longer instances, with sentence lengths above 100, where ON-BERT improves by 6 percentage points over BERT-GT. This proves that the convolutional neural network is very effective in modeling the interactions between complex long sentences and can significantly improve the extraction results. It also fully demonstrates that the models in this chapter are competitive and perform well in different sentence lengths.

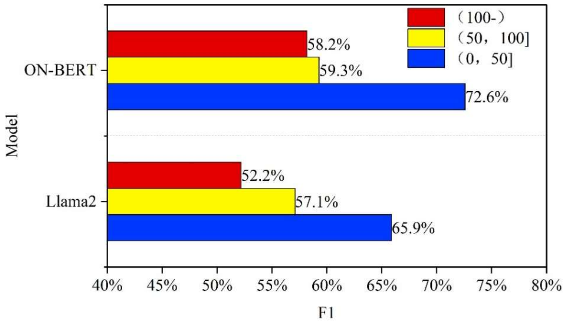
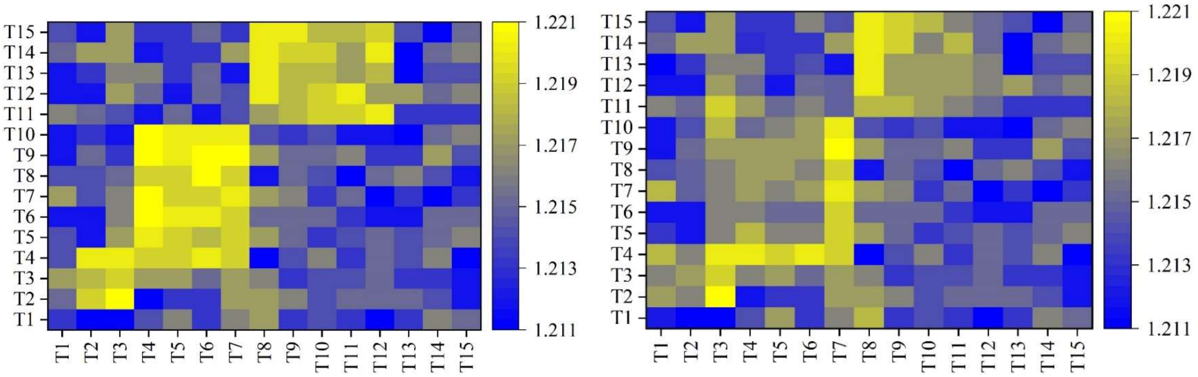


Figure 4: Experiment results of different sentence lengths

III. C. 5) Case studies

The method proposed in this chapter uses the attention mechanism to construct a weight map for a given text, in order to further explore the effect of the weight map after convolution, this subsection visualizes the weight map obtained by the model by drawing heat maps, and the results of the drawing are shown in Fig. 5, with (a) and (b) indicating the performance of ON-BERT and Llama2, respectively. These two heat maps show the different weights between the 15 Token respectively, and the difference of the weight maps can be found more intuitively by the darkness of the colors in the heat maps. From Figure (a), it can be observed that ON-BERT is able to infer the relationship from the target entity to most of the other legal texts, while the Llama2 model in Figure (b) does not have a particularly heavy weight share between these two entities, indicating that the relationship between the entities cannot be clearly recognized. The case study further confirms the usefulness of convolutional neural networks.



(a) ON-BERT

(b) Llama2

Figure 5: Heat maps of the two weight graphs

IV. Conclusion

The ON-BERT model in this study performs well in the legal text entity relationship extraction task. By introducing hierarchical parsing and graph convolutional networks, ON-BERT significantly outperforms the traditional model in terms of accuracy, recall and F1 value. In the experiments, the accuracy of ON-BERT is 82.55%, and the F1 value reaches 83.56%, which is 3.92% higher than that of the BERT model and 4.0% higher than that of the BERT model. Especially when dealing with the relationship between “Has_Crime” and “Has_Sentence_Result”, the accuracy rate is more than 90%, which shows the superiority of the model in the text of criminal cases. In addition, the ON-BERT model not only improves the extraction accuracy, but also has a significant advantage in inference speed and training efficiency, which reduces the training time by about 4 times compared with the baseline model, and the inference time is also greatly shortened. The experimental results show that the ON-BERT model is able to deal with long-distance dependencies in complex texts while ensuring high efficiency, which fully demonstrates the potential of deep learning application in legal text processing.

References

[1] Hurka, S., & Steinebach, Y. (2021). Legal instrument choice in the European Union. JCMS: Journal of Common Market Studies, 59(2), 278-296.

[2] Papagianneas, S. (2024). Smart courts, smart ustice? automation and digitisation of courts in China. Asian Journal of Law and Society, 1, 27.

[3] Zhang, D., Lu, T., Zhang, W., & Yang, C. (2023). A Neural Network Method for Systematic Evaluation of Informatization Development Level in Smart Court Construction. Journal of Internet Technology, 24(4), 915-922.

[4] Peng, J., & Xiang, W. (2019). The rise of smart courts in China: opportunities and challenges to the judiciary in a digital age. Nordic J Law Soc Res (NNJLSR), 9, 345-372.

[5] Bhattacharya, P., Ghosh, K., Pal, A., & Ghosh, S. (2022). Legal case document similarity: You need both network and text. Information Processing & Management, 59(6), 103069.

[6] Wagh, R. S., & Anand, D. (2020). Legal document similarity: a multi-criteria decision-making perspective. PeerJ Computer Science, 6, e262.

[7] Zhou, Y., Liu, L., Chen, Y., Huang, R., Qin, Y., & Lin, C. (2024). A novel MRC framework for evidence extracts in judgment documents. Artificial Intelligence and Law, 32(1), 147-163.

[8] Dhanani, J., Mehta, R., & Rana, D. (2021). Legal document recommendation system: A cluster based pairwise similarity computation. Journal of Intelligent & Fuzzy Systems, 41(5), 5497-5509.

[9] Bellandi, V., Bernasconi, C., Lodi, F., Palmonari, M., Pozzi, R., Ripamonti, M., & Siccardi, S. (2024). An entity-centric approach to manage court judgments based on Natural Language Processing. Computer Law & Security Review, 52, 105904.

[10] Chen, Y., Sun, Y., Yang, Z., & Lin, H. (2020, December). Joint entity and relation extraction for legal documents with legal feature enhancement. In Proceedings of the 28th international conference on computational linguistics (pp. 1561-1571).

[11] Huang, W., Hu, D., Deng, Z., & Nie, J. (2020). Named entity recognition for Chinese judgment documents based on BiLSTM and CRF. EURASIP Journal on Image and Video Processing, 2020, 1-14.

[12] Ji, D., Tao, P., Fei, H., & Ren, Y. (2020). An end-to-end joint model for evidence information extraction from court record document. Information Processing & Management, 57(6), 102305.

[13] Zhang, H., Guo, J., Wang, Y., Zhang, Z., & Zhao, H. (2023). Judicial nested named entity recognition method with MRC framework. International Journal of Cognitive Computing in Engineering, 4, 118-126.

[14] Li Siyi,Zhang Mingrui & Piggott Matthew D.. (2023). End-to-end wind turbine wake modelling with deep graph representation learning. Applied Energy,339,

[15] Chuhan Gao,Guixian Xu & Yueting Meng. (2024). Integrated Extraction of Entities and Relations via Attentive Graph Convolutional Networks. Electronics,13(22),4373-4373.

[16] Jiao Han & Kang Jia. (2024). Entity relation joint extraction method for manufacturing industry knowledge data based on improved BERT algorithm. Cluster Computing,27(6),7941-7954.