IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 16, 2019, accepted October 1, 2019, date of publication October 7, 2019, date of current version October 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2945771

MANN: A Multichannel Attentive Neural Network for Legal Judgment Prediction

SHANG LI[1], HONGLI ZHANG[1], LIN YE[1], XIAODING GUO[1], AND BINXING FANG[1,2]
[1]School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China
[2]Institute of Electronic and Information Engineering of UESTC in Guangdong, Dongguan 523808, China

Corresponding author: Hongli Zhang (zhanghongli@hit.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFC0830903 and Grant 2018YFC0830602, and in part by the National Natural Science Foundation of China (NSFC) under Grant 61872111 and Grant 61732022.

ABSTRACT Recent years have witnessed an opportunity to improve trial efficiency and quality by predictive analysis of massive judgment documents. A practical legal judgment prediction (LJP) system should provide a judge with feasible judgment suggestions, including the charges, applicable law articles, and prison term, whereas most existing works focus on only part of the LJP task. Inspired by the impressive success of deep neural networks in a wide range of application scenarios, we propose a multichannel attentive neural network model, MANN, which learns from previous judgment documents and performs the integrated LJP task in a unified framework. In general, MANN takes the textual description of a criminal case as the input for attention-based neural networks to learn its latent feature representations oriented to the case fact, the defendant persona, and relevant law articles. Moreover, we adopt a two-tier structure to empower attentive sequence encoders to hierarchically model the semantic interactions from different parts of case description at both the word and sentence levels. The experiments are conducted on four real-world datasets of criminal cases in mainland China. The experimental results demonstrate that MANN achieves state-of-the-art LJP performance on all evaluation metrics.

INDEX TERMS Legal intelligence, judgment prediction, neural networks, attention mechanism.

I. INTRODUCTION

In recent years, legal judgment prediction (LJP) has become a research hotspot in legal intelligence (i.e., the application of artificial intelligence techniques in the field of law). It is a promising technique that aims to provide appropriate judgment advice, including the charges, applicable law articles, and prison term. LJP plays an important role in legal assistant systems, which can help legal professionals (e.g., judges, lawyers, and prosecutors) to improve their work efficiency and reduce the risk of making mistakes. Furthermore, it can benefit ordinary people who lack rich legal knowledge but desire to know the possible judgment result by describing a case they are concerned about. By exploiting the legal knowledge contained in massive law articles and case judgment documents, LJP will free people from the laborious tasks of information retrieval and data analysis.

However, it is not trivial to train an intelligent machine judge to predict appropriate judgment results due to the complexity of judicial trials. In civil law jurisdictions, e.g., mainland China, human judges deal with legal cases via comprehensive consideration of the case facts, personal details of the defendant, and statutory laws, rather than with reference to decisions of precedent cases. As shown in Fig. 1, a judgment document of a criminal case in China always includes the defendant persona, case facts, and judgment decision. This legal institution endows judges with the discretion to make a final decision in a case-by-case paradigm, where any specific circumstances should be taken into account. As far as we can see, LJP in real-world scenarios is still confronted with two main challenges: (1) different parts of the textual description may contain latent features well representing a specific case from different perspectives, whereas not all the sentences or words are informative enough to extract these feature representations; (2) human judges execute multiple subtasks of legal judgment as a whole, thus, learning to perform a complete LJP task according to human logic will improve the credibility and interpretability.

The majority of existing works attempt to resolve the judgment prediction task by formalizing it as a classification problem. Early efforts either employed off-the-shelf classification models [1]–[3] with shallow features extracted from case

The associate editor coordinating the review of this manuscript and approving it for publication was Yu-Huei Cheng.

被告人冯某某。曾因犯盗窃罪，于2015年7月31日被判处有期徒刑十个月，罚金人民币一千元，2016年1月3日刑满释放。	Defendant Persona	**The defendant Feng was sentenced** to 10 months imprisonment and a fine of RMB 1,000 on July 31, 2015 for a crime of theft. He was released on January 3, 2016.
公诉机关指控，2016年5月20日，被告人冯某某在北京市石景山区鲁谷东路XX商店内，盗窃被害人肖某人民币1337元。 2016年6月8日，被告人冯某某伙同他人，在北京市海淀区北洼路XX饭馆内，盗窃被害人郭某XX牌电动自行车1辆。	Case Facts	**The public prosecutor accused that**, on May 20, 2016, the defendant Feng stole RMB 1,337 from the victim Xiao in the XX store, East Lugu Road, Shijingshan District. On June 8, 2016, the defendant Feng and others stole a XX electric bicycle from the victim Guo in the XX restaurant, Beiwa Road, Haidian District.
本院认为，被告人冯某某多次盗窃他人财物，数额较大，其行为已构成盗窃罪…鉴于被告人冯某某曾因故意犯罪被判处有期徒刑，系累犯，本院依法对其从重处罚…依照《中华人民共和国刑法》第二百六十四条、第六十五条第一款、第六十七条第三款、第五十三条第一款、第六十四条之规定，判决如下：一、被告人冯某某犯盗窃罪，判处有期徒刑一年二个月，罚金人民币五千元…	Judgment Decision	The court hold that, the defendant Feng has stolen others for many times, and his behavior has constituted the crime of theft. As the defendant Feng was sentenced to a fixed-term imprisonment, he is a recidivist and should be punished more heavily . In accordance with the Article 264… of the Criminal Law of the People's Republic of China, **the decisions are as follows**: 1. The defendant Feng committed the crime of theft and was sentenced to a fixed-term imprisonment of one year and two months and a fine of RMB 5,000…

FIGURE 1. An example judgment document of a criminal case in China (original Chinese text and its English translation).

text [4], [5] or case profiles [6] or attained deeper semantic understanding of case descriptions by manually annotating cases and designing discriminative features [7]. These approaches are time-consuming and hard to scale due to relying heavily on expert knowledge and human annotation. Despite the introduction of artificial intelligence and natural language processing (NLP) methods that can advance the related tasks in the legal intelligence field, such as charge prediction [8], legal reading comprehension [9], and court view generation [10], the ability to learn sufficient semantic representations from different parts of case description remains unsolved. For example, the defendant persona does not receive sufficient attention despite its nonnegligible effect on the precise adjustment of prison terms and application of law articles on a case-by-case basis.

To address these problems, we propose a practical LJP approach by exploiting recent advances in deep neural networks. Specifically, we employ Bidirectional Gated Recurrent Units (Bi-GRU) to construct hierarchical sequence encoders to learn semantic representations from different parts of case description at the both word and sentence levels. Inspired by the successful applications of neural networks and an attention mechanism in document classification [11], neural machine translation [12] and adaptation model [13], we put forward a multichannel attentive neural network model, MANN, which consists of fact-channel, article-channel, and persona-channel sequence encoders integrated with context attention vectors. On one hand, each channel focuses on selecting informative words and sentences in its target part of case description. On the other hand, a dynamic mechanism is adopted to generate context attention vectors under the guidance of other channels, since law article embedding should depend on its own representation as well as absorb related information from the fact and persona channels. With the case description embeddings from multiple channels, we jointly perform the integrated LJP task and construct specific predictors for the charges, law articles, and prison term.

To investigate the advancement of our approach in the LJP task, we conduct experiments on four real-world datasets containing large-scale criminal cases in mainland China. The experimental results demonstrate that the proposed MANN achieves state-of-the-art LJP performance on all evaluation metrics.

Our contributions in this paper are summarized as follows:

(1) To the best of our knowledge, this is the first study that investigates the entire LJP task in the context of complete case description, including both the case fact and defendant persona, which will undoubtedly contribute to a better LJP performance compared to the existing methods, as well as further researchers' understanding of the LJP task in real-world scenarios.

(2) We propose an innovative framework, MANN, to jointly carry out different parts of the LJP task in a case-by-case paradigm like human judges. The hierarchical sequence encoder integrated with a multichannel attention mechanism is beneficial for learning better representations from informative case descriptions.

(3) We perform a series of experiments on four real-world datasets of Chinese criminal cases. The results clearly show the improvement of our approach over all baselines on the LJP task.

The rest of this paper is organized as follows. Section II briefly reviews the related work. The formalization of the LJP task is described in Section III. In Section IV, we propose the overall MANN framework and detailed methods. The experimental results and analyses are presented in Section V. Finally, Section VI contains the concluding remarks.

II. RELATED WORK

The LJP problem has drawn increasing attention from the research community in recent years. Relevant issues in the field of legal intelligence have also been studied.

In early studies on LJP, most researchers only focused on certain subtasks and tended to formalize them as a text classification problem. Hachey and Grover [1] exploited off-the-shelf machine learning models to identify summary-worthy legal sentences for automatic court rulings. The work of Gonçalves and Quaresma [2] aimed to classify legal text in 3,000 categories based on a taxonomy of legal concepts. Liu *et al.* [4] proposed a case-based reasoning system and adopted a KNN model to classify 12 common criminal charges. Katz *et al.* [6] built randomized trees with features extracted from case profiles to predict the US Supreme

Court's behavior. The work in [7] applied machine learning methods to identify robbery and intimidation cases and predict the sentence based on manually defined 21 legal factor labels. More recently, the work of Aletras *et al.* [14] aimed to predict decisions of the European Court of Human Rights by training Support Vector Machine (SVM) binary classifiers with textual features, such as N-grams and topics. Similarly, Sulea *et al.* [15], [16] used a linear SVM classifier to predict law area and case judgments of the French Supreme Court. Boella *et al.* [17] used term frequency-inverse document frequency (TF-IDF) and information gain for feature selection and then built an SVM classifier to identify the relevant domain to which the given legal text belongs. Liu and Chen [18] used an SVM algorithm to classify the judgment text according to relevant law articles, sentiment analysis of crime facts and prison term. Although these efforts take full advantage of supervised learning method, they suffer from the scalability problem due to relying heavily on feature design and manual annotation. Our approach, however, employs an attention-based neural network to learn comprehensive representation of a case without human efforts to design and annotate specific features.

In addition to the charge prediction, some researchers have explored the method of identifying applicable law articles for a given legal case. Liu and Hsieh [5], Liu and Liao [19] proposed an intuitive solution of converting the multilabel classification problem into a multiclass classification problem by focusing on a fixed set of article combinations. Despite the satisfactory results they have obtained in the classification of larceny and gambling crimes, this approach is hardly applied extensively to real scenarios, where the number of candidate law articles can be very large. To solve the scalability problem, the work in [20] reported a two-step strategy consisting of preliminary article classification by SVM and reranking the results using word-level features and cooccurrence tendency among law articles. This inspired us to roughly filter out uncorrelated law articles instead of feeding all the articles into the sequence encoder, which can contribute to more refined article embedding for a certain case.

Motivated by the successful application of deep learning methods in NLP tasks, researchers proposed introducing neural network models into the field of legal intelligence. Chalkidis and Androutsopoulos [21] reported their achievement in contract element extraction and employed Bidirectional Long Short-Term Memory (Bi-LSTM) operating on word, part-of-speech (POS) tag, and token-shape embeddings without any manually written rules. Wei *et al.* [22] used a Convolution Neural Network (CNN) to implement text classification for legal document review, and their experimental results show that the CNN model performs much better than SVM with a larger volume training dataset. Luo *et al.* [8] incorporated an attention mechanism into the stacked neural network to predict charges with legal basis, which is reasonably generalizable. Zhong *et al.* [23] proposed a multi-task learning framework to incorporate the topological dependencies of multiple subtasks into judgment prediction, but they neglected the interactions of subtask results. Ye *et al.* [10] formulated the court view generation task as a text-to-text natural language generation (NLG) problem and presented a label-conditioned sequence-to-sequence model with attention to generate charge-discriminative court views. However, these studies only focus on the analysis of fact description without paying attention to the defendant persona, which will affect the precise adjustment of prison term and application of law articles on a case-by-case basis.

In summary, previous studies have advanced several aspects of the LJP task. Nevertheless, it remains a challenge to comprehensively learn sufficient semantic representations from multiple parts of case description and carry out the entire LJP task in a unified framework. This is why MANN is introduced in this study.

III. LEGAL JUDGMENT PREDICTION

In this paper, we focus on the LJP problem in the context of judging criminal cases in mainland China, which is one of the civil law jurisdictions. As the core data driving our research, the judgment document D of a criminal case can be depicted as:

$$D = [D_p, D_f, D_r], \tag{1}$$

where

- D_p: the defendant persona, including the physiological features that determine the criminal liability (e.g., age, health condition, mental status), criminal records, etc.;
- D_f: the fact description of events that happened in the case, such as crime acts, crime locations, crime consequences, crime tools and other articles involved in the case, etc.;
- D_r: the judgment decision from the court or judge, which consists of three main aspects, i.e., the charges R_c, applicable law articles R_a, and prison term R_t.

The complete D can only be acquired in case of the accomplishment of legal judgment, which means the input of a practical LJP task only includes personal details about the defendant and the fact description, i.e., $[D_p, D_f]$, and the targeted output is $[R_c, R_a, R_t]$. Therefore, LJP can be formalized as the following five-tuple task:

$$[D_p, D_f, R_c, R_a, R_t]. \tag{2}$$

Given $[D_p, D_f]$, the task of LJP aims to find $[\hat{R}_c, \hat{R}_a, \hat{R}_t]$ as follows:

$$\begin{cases} \hat{R}_c = \arg\max P(R_c|[D_p, D_f]), & (3) \\ \hat{R}_a = \arg\max P(R_a|[D_p, D_f]), & (4) \\ \hat{R}_t = \arg\max P(R_t|[D_p, D_f]), & (5) \end{cases}$$

where $P(R_i|[D_p, D_f])(i \in \{c, a, t\})$ is the probability of the predicted judgment decision.

To sum up, the innovation of our definition for the LJP task is mainly reflected as follows:

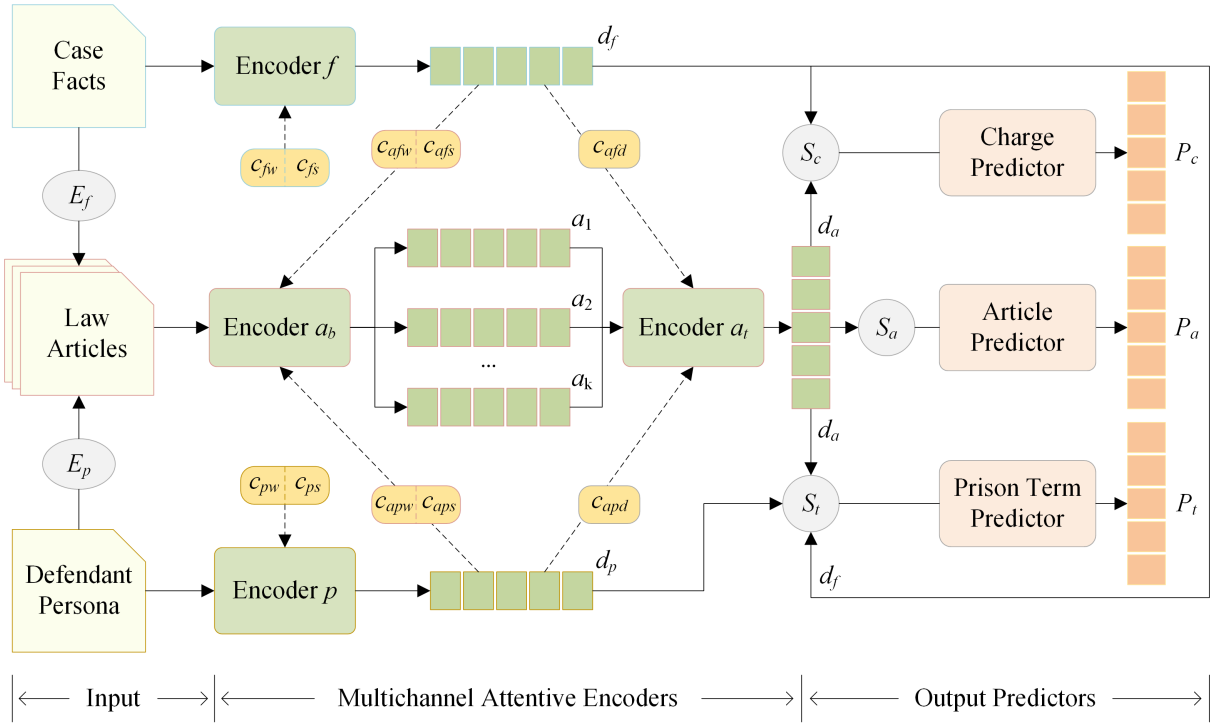

FIGURE 2. An overview of the MANN framework.

(1) Compared with previous works that focused either on charge prediction or law article extraction, our LJP encompasses charge prediction, law article prediction, and prison term prediction into an integrated task, which is supported by the proposed MANN framework.

(2) By taking the defendant persona into account, LJP will be empowered with the ability to make a refined prediction of law articles and prison term based on more sufficient case-specific semantic representation.

IV. METHOD

To carry out the LJP task, we propose a multichannel attentive neural network (MANN) model. As shown in Fig. 2, the MANN framework consists of the following parts:

(1) **Input layer**. The fact description and defendant persona are first fed into law article extractors E_f and E_p to generate two candidate sets of relevant law articles, along with which transformed into distributed representation as the inputs of subsequent sequence encoders.

(2) **Multichannel attentive encoders**. The above input is then passed to three hierarchical encoders sharing similar structures with a word sequence encoder and a sentence sequence encoder. By incorporating them with word-level and sentence-level attention context vectors, i.e., c_{fw} and c_{fs}, the fact-channel encoder f is capable of capturing informative words and sentences and generates the fact embedding d_f. Analogously, the persona embedding d_p is generated by the persona-channel encoder p with attention context vectors c_{pw} and c_{ps}. For law articles, the article-channel bottom encoder a_b first embeds candidate law articles into a sequence of article embeddings $[a_1, a_2, \ldots, a_k]$ using context vectors c_{afw} and c_{afs}, c_{apw} and c_{aps} dynamically generated from d_f and d_p, respectively. Given the embedding sequence of candidate law articles $[a_1, a_2, \ldots, a_k]$, the article-channel top encoder a_t produces the final article embedding d_a by leveraging context vectors c_{afd} and c_{apd} dynamically generated from d_f and d_p, respectively.

(3) **Output Predictors**. The article predictor aims to decide the most relevant law articles that can support the judgment of the input case and outputs the predicted article distribution P_a by passing the article embedding d_a into a softmax function. The concatenations $[d_f, d_a]$ and $[d_f, d_p, d_a]$ are also passed to a softmax classifier to generate the predicted charge distribution P_c and prison term distribution P_t, respectively.

A. INPUT LAYER

1) CANDIDATE LAW ARTICLE EXTRACTION

In civil law jurisdictions, the judgment result of a case is decided jointly by the specific circumstance and relevant law articles. The articles in the Criminal Law of the People's Republic of China mainly fall into two types: (1) basic articles, which define the common rules that must be abided by in the determination of crime, identification of responsibility and application of penalty; (2) auxiliary articles, which demonstrate some particular conditions of the involved case and defendants that should be taken into account on the basis of the application of basic articles.

In terms of the LJP task, basic articles are crucial for the determination of charges and guideline range of prison term by the interpretation of "*What did the defendant principally do?*" according to the case fact description, whereas auxiliary articles can help the subtle calibration of prison term by the analysis of "*What special circumstances did the defendants fit?*" based on the case fact description and the defendant persona.

Due to the large number of law articles, predicting relevant articles for a legal case can be formalized as a multilabel classification problem. However, benefiting from the fact that a certain case is theme-related to only a small fraction of law articles, we can downsize the candidate set of relevant articles fed into the subsequent neural networks for deep semantic comprehension. Inspired by [8], [20], we carry out the extraction of candidate law articles by transforming it into a multiple binary classification with high efficiency and scalability. We employ word-based SVM to build the binary classifier for each law article, whereby the output of each classifier represents the relevance of the law article to the given case. Specifically, word-level TF-IDF features and linear kernel are used for binary classification.

To evaluate the SVM law article extractor, we use a parameter k to control the extraction number of candidate law articles and calculate the recall rate of top k extraction. Taking the annotated applicable law articles of 100,000 cases randomly selected from our real-world dataset as reference, the SVM law article extractor achieves 0.883, 0.935, 0.948, 0.957 recall regarding the top 10, 20, 30 and 50 extracted law articles, respectively. With the top 20 candidate law articles, the SVM law article extractor can obtain a recall of over 0.93, which is accurate enough for the subsequent refined law article prediction.

2) TEXT PREPROCESSING AND WORD EMBEDDING

Since all the judgment documents are written in Chinese, word segmentation is carried out first. To avoid possible interference with the subsequent process of document embedding, some insignificant words (e.g., names of people, places, organizations) are filtered by employing POS tagging and named entity recognition technology.

After text preprocessing, the case description is transformed into a word sequence with end-of-sentence tags. To make these Chinese words calculable, each word must be mapped into a vector space through the distributed representation process [24]. In this paper, we use word2vec and the CBOW (Continuous Bag-of-Words) model optimized by a negative sampling technique to complete the distributed representation of text and map all words in the text into the same vector space.

B. MULTICHANNEL ATTENTIVE ENCODERS

1) HIERARCHICAL SEQUENCE ENCODER

Intuitively, a judgment document has a hierarchical structure [11], i.e., a document can be represented as a sequence of sentences, and a sentence is a sequence of words. Thus, we adopt a two-tier structure to construct the document representation with a word sequence encoder and a sentence sequence encoder, as shown in Fig. 3.

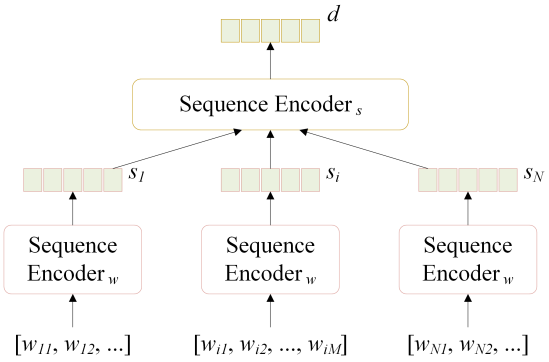

FIGURE 3. Hierarchical Sequence Encoder.

Suppose that a judgment document d contains N sentences $s_i (i \in [1, N])$, where s_i consists of M words and $w_{ij} (j \in [1, M])$ is the jth word in the ith sentence; then, the document embedding d can be represented as:

$$d = f([s_1, s_2, \ldots, s_N]), \tag{6}$$

$$s_i = g([w_{i1}, w_{i2}, \ldots, w_{iM}]), \tag{7}$$

where g and f are the word encoder and the sentence encoder, respectively. In this paper, we employ Bi-GRU to implement the two isostructural encoders.

GRU Network Recurrent Neural Network (RNN) 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. Typical RNNs include the traditional RNN, LSTM, GRU and their variants. A common LSTM unit [25] 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. GRU is a variant of LSTM whose unit structure [26] is similar to LSTM but simpler, as shown in Fig. 4. Compared to LSTM, GRU removes the memory cell 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:

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

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

$$\hat{h}_t = \tanh(W_n x_t + U_n(r_t \odot h_{t-1}) + b_n), \tag{10}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \hat{h}_t, \tag{11}$$

where r_t means the result of the reset gate, z_t means the result of the update gate, and $\hat{h}_t$ is the intermediate state when calculating the hidden state h_t. The operator $\odot$ indicates the elementwise multiplication of two matrices. As shown

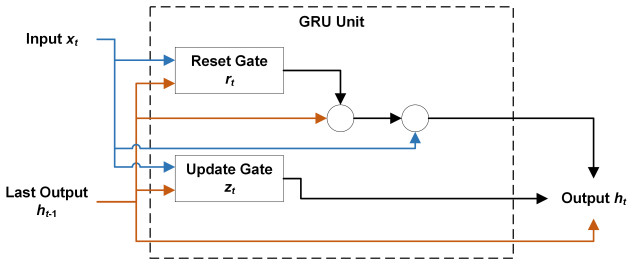

FIGURE 4. Structure of the GRU unit.

in (12), the tanh activation function resembles the identity function more closely, which ensures the neural network to learn efficiently when its weights are initialized with small random values. Specifically, the tanh function maps a distribution around zero to another one and has the zero-centered output with the range of (-1, 1). This property will not only avoid the zigzag phenomenon, but also facilitate the learning of neurons in the next layer.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{12}$$

Bi-GRU Sequence Encoder Bi-GRU predicts or labels each element of the sequence based on the element's past and future contexts by concatenating the states of two GRUs, with one processing the sequence forward and the other one backward. Given a sequence $[x_1, x_2, ldots, x_T]$, where $x_t (t \in [1, T])$ is the embedding vector of the input element t, the forward GRU encodes the sequence from x_1 to x_T, while the backward GRU encodes the sequence from x_T to x_1. For the element x_t, the forward hidden state h_{ft} and the backward hidden state h_{ft} can be produced as:

$$h_{ft} = GRU([x_1, x_2, ldots, x_t]), \tag{13}$$
$$h_{bt} = GRU([x_T, x_{T-1}, ldots, x_t]), \tag{14}$$

then we can obtain the Bi-GRU hidden state for x_t by concatenating h_{ft} and h_{ft}:

$$h_t = [h_{ft}, h_{bt}]. \tag{15}$$

2) MULTICHANNEL ATTENTION MECHANISM

In a judgment document, not all sentences contribute equally to the representation of case details; likewise, not all words are essential to represent the key meaning of a sentence. Hence, the sequence encoding calls for an approach that can distinguish informative elements from insignificant ones in the sequence of words or sentences. The work of [11], [12] inspired us to incorporate an attention mechanism with the proposed hierarchical sequence encoders. As shown in Fig. 5, we introduce the context vectors c_w and c_s to attentively aggregate the representation of informative words and sentences to generate sentence-level and document-level vectors, respectively.

Given an input sequence of word annotations $[h_{i1}, h_{i2}, \ldots, h_{iM}]$, we can first obtain the Bi-GRU hidden

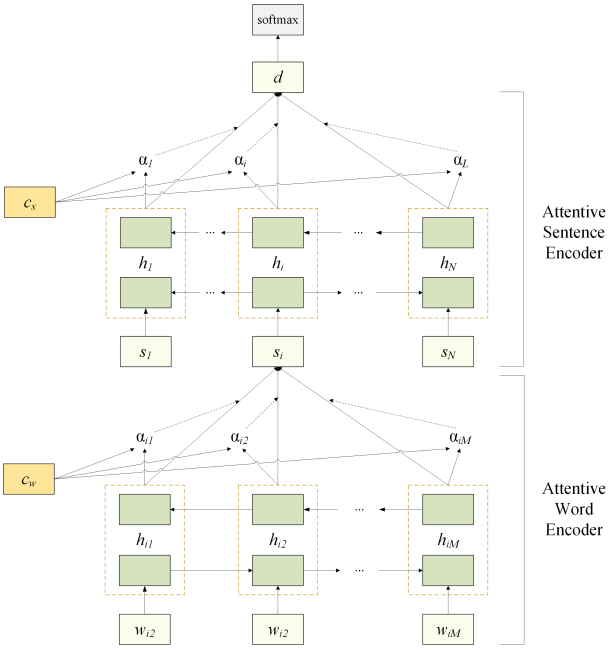

FIGURE 5. Hierarchical Attention Network.

state of h_{ij} through a one-layer Multilayer Perception (MLP) as

$$u_{ij} = \tanh(W_w h_{ij} + b_w). \tag{16}$$

Then, we measure how well u_{ij} and the context vector c_w match by their cosine similarity and obtain a normalized weight α_{ij} for the annotation h_{ij} through a softmax function as

$$\alpha_{ij} = \frac{\exp(u_{ij}^T c_w)}{\sum_j \exp(u_{ij}^T c_w)}. \tag{17}$$

Afterwards, the sentence-level vector s_i is calculated as a weighted sum of word annotations h_{ij}:

$$s_i = \sum_j \alpha_{ij} h_{ij}. \tag{18}$$

Similarly, the document-level vector d is calculated as the following:

$$u_i = \tanh(W_s h_i + b_s), \tag{19}$$
$$\alpha_i = \frac{\exp(u_i^T c_s)}{\sum_i \exp(u_i^T c_s)}, \tag{20}$$
$$d = \sum_i \alpha_i h_i. \tag{21}$$

Note that c_w and c_s are global context vectors in [12], where they are randomly initialized and jointly learned during the training process. In our work, we follow this approach for building the fact-channel encoder f and the persona-channel encoder p. Specifically, we employ the word-level attention context vector c_{fw} and sentence-level attention context vector c_{fs} to produce the fact embedding d_f based on informative

words and sentences, while the persona-channel encoder p generates the persona embedding d_p by incorporating the context vectors c_{pw} and c_{ps}.

Generally, this setting is appropriate for the embedding of independent (part of) documents that have no content dependency on others, such as the fact description and defendant persona in a judgment document. However, in the scenario of the LJP task, a judge can decide which law articles are most supportive for a specific case only if given the fact description and defendant persona. This results in the dependency of the article-channel encoder on both the fact-channel encoder f and persona-channel encoder p.

Since one law article may consist of several sentences, the article-channel encoder can also be built with the similar hierarchical architecture as the fact-channel encoder f and persona-channel encoder p, as shown in Fig. 3. However, to produce a sequence of article embeddings $[a_1, a_2, \ldots, a_k]$ that can capture informative words and sentences closely related to the specific case fact and defendant persona, we first construct an article-channel bottom encoder a_b that uses word-level and sentence-level attention context vectors dynamically generated under the guidance of the fact embedding d_f and persona embedding d_p. The involved context vectors are calculated as follows:

$$c_{afw} = W_{fw} d_f + b_{fw}, \tag{22}$$
$$c_{afs} = W_{fs} d_f + b_{fs}, \tag{23}$$
$$c_{apw} = W_{pw} d_p + b_{pw}, \tag{24}$$
$$c_{aps} = W_{ps} d_p + b_{ps}, \tag{25}$$

where c_{afw} and c_{afs} are the word-level and sentence-level context vectors focusing on the relevance of the law article to the fact description, c_{apw} and c_{aps} are guided by the defendant persona, and W_* and b_* are the weight matrix and bias vector, respectively.

Given the embedding sequence of candidate law articles $[a_1, a_2, \ldots, a_k]$ generated by the article-channel bottom encoder a_b, we construct an article-channel top encoder a_t to obtain a more refined embedding at the article level by attentively selecting law articles with high relevance to the fact and defendant persona of the input case. Specifically, we leverage document-level context vectors c_{afd} and c_{apd} to guide the fact-channel and persona-channel attention for the final article embedding. Similarly, c_{afd} and c_{apd} are dynamically generated with regard to the fact embedding d_f and persona embedding d_p as follows:

$$c_{afd} = W_{fd} d_f + b_{fd}, \tag{26}$$
$$c_{apd} = W_{pd} d_p + b_{pd}, \tag{27}$$

where W_* and b_* are the weight matrix and bias vector, respectively.

C. OUTPUT PREDICTORS

With the fact embedding d_f, the persona embedding d_p and the article embedding d_a of an input case, we aim to predict its judgment results and construct three specific predictors for the three parts of the LJP task described in Section III.

1) ARTICLE PREDICTOR

The article predictor aims to select the most relevant law articles that can support the judgment of the input case. With the article embedding vector d_a, we apply a softmax function to obtain the predicted article probability distribution $P_a = [p_{a1}, p_{a2}, \ldots, p_{aK}]$, each $p_{ak} \in [0, 1]$ represents the probability of article k being applicable for the input case, where $k \in [1, K]$ and K is the number of distinct law articles in the dataset.

Given a threshold τ_a, we take all the articles with a probability higher than τ_a as positive predictions and ultimately obtain the final article prediction results $\hat{R}_a = [\hat{r}_{a1}, \hat{r}_{a2}, \ldots, \hat{r}_{aK}]$, where $\hat{r}_{ak}$ is the prediction result on article k and is computed as:

$$\hat{r}_{ak} = \begin{cases} 1 & p_{ak} \geq \tau_a, \\ 0 & p_{ak} < \tau_a. \end{cases} \tag{28}$$

The training objective for the article predictor is to minimize the cross-entropy between the predicted article probability distribution P_a and the ground-truth distribution R_a; thus, the article prediction loss is calculated as:

$$Loss_a = -\sum_{k=1}^{K} r_{ak} log(p_{ak}), \tag{29}$$

where r_{ak} and p_{ak} are the ground-truth and predicted probability of article k for the input case, respectively. The ground-truth article distribution R_a is generated by setting $r_{ak} = \frac{1}{t_a}$ for positive labels and $r_{ak} = 0$ for negative labels, where t_a is the number of positive labels.

2) CHARGE PREDICTOR

The charge predictor aims to determine applicable charges for the input case by considering both the fact description and relevant law articles. Therefore, we first concatenate the fact embedding d_f and the article embedding d_a into an integrated vector and then pass it to a softmax classifier to obtain the predicted charge probability distribution $P_c = [p_{c1}, p_{c2}, \ldots, p_{cG}]$, each $p_{cg} \in [0, 1]$ represents the probability of charge g being applicable for the input case, where $g \in [1, G]$ and G is the number of distinct charges in the dataset.

Similar to the article predictor, we set a threshold τ_c to output charges with the probability higher than τ_c as positive predictions, and obtain the charge prediction results $\hat{R}_c = [\hat{r}_{c1}, \hat{r}_{c2}, \ldots, \hat{r}_{cG}]$, where $\hat{r}_{cg} \in \{0, 1\}$ is the prediction result on charge g. We also use the cross-entropy to measure the charge prediction loss as:

$$Loss_c = -\sum_{g=1}^{G} r_{cg} log(p_{cg}), \tag{30}$$

where r_{cg} and p_{cg} are the ground-truth and predicted probability of charge g for the input case, respectively.

The ground-truth charge distribution R_c is generated by setting $r_{cg} = \frac{1}{t_c}$ for positive labels and $r_{cg} = 0$ for negative labels, where t_c is the number of positive labels.

3) PRISON TERM PREDICTOR

The prison term predictor aims to produce a reasonable prison term interval based on the comprehensive consideration of the fact description, relevant law articles, and defendant persona. By passing the concatenation of the fact embedding d_f, the article embedding d_a, and the persona embedding d_p to a softmax classifier, we obtain the predicted prison term interval probability distribution $P_t = [p_{t1}, p_{t2}, \ldots, p_{tH}]$, each $p_{th} \in [0, 1]$ represents the probability of prison term interval h being applicable for the input case, where $h \in [1, H]$ and H is the number of nonoverlapping intervals in the dataset.

Again, with a threshold τ_t, we can obtain the prison term prediction results $\hat{R}_t = [\hat{r}_{t1}, \hat{r}_{t2}, \ldots, \hat{r}_{tH}]$, where $\hat{r}_{th} \in \{0, 1\}$ is the prediction result on prison term interval h. The cross-entropy is used to calculate the prison term prediction loss as:

$$Loss_t = -\sum_{h=1}^{H} r_{th} log(p_{th}), \tag{31}$$

where r_{th} and p_{th} are the ground-truth and predicted probability of prison term interval h for the input case, respectively. The ground-truth prison term interval distribution R_t is generated by setting $r_{th} = 1$ for the positive label and $r_{th} = 0$ for negative labels.

4) TRAINING

Considering the three training objectives as a whole, we use a weighted sum of $Loss_c$, $Loss_a$ and $Loss_t$ as the overall loss function:

$$Loss = \alpha * Loss_c + \beta * Loss_a + \gamma * Loss_t \tag{32}$$

where α, β, and γ are hyperparameters used to control the weight of these three parts in the loss function. In practice, we set all of α, β, and γ as 1, under the assumption that the three parts of the LJP task are of equal importance.

V. EXPERIMENTS

To evaluate the proposed MANN framework, we conduct a series of experiments on the LJP task over four real-world datasets of criminal cases in mainland China; further, several baselins are employed to compare with our approach.

A. DATASETS

We construct the first dataset on the basis of the Chinese AI and Law challenge (CAIL2018) dataset [27]. Each case in CAIL2018 is collected from China Judgments Online[1] and consists of the fact description and judgment results, i.e.,charges, law articles, and prison term.

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

TABLE 1. Statistics of the four datasets.

Datasets	RCAIL	CJO-S	CJO-L	PKU
Training Set	1,094,124	80,000	160,000	160,000
Validation Set	136,765	10,000	20,000	20,000
Testing Set	136,765	10,000	20,000	20,000
Charges	58	10	10	20
Law Articles	68	15	15	28
Prison Term Intervals	11	11	11	11

Although CAIL2018 contains a large volume of criminal cases, we find considerable annotation errors (e.g., the inconsistency between the charges and articles) of judgment results by manual check. Moreover, there are some infrequent charges and law articles, such as Cultural *relic scalping*, *defamation*, and *perjury*. Thus, we filter out the cases containing either mislabeled judgment results or infrequent charges and law articles and ultimately construct a refined CAIL2018 dataset, RCAIL, which consists of 1,367,654 cases with 58 distinct charges, 68 distinct law articles, and 11 nonoverlapping prison term intervals. It is worth noting that, the distribution of different categories in RCAIL is still quite imbalanced. For example, the top 10 charges cover 82.1% of all the cases, whereas the bottom 10 charges cover only 1.5%.

Despite a large number of cases, RCAIL has a major deficiency that it includes only the fact description of a criminal case and neglects the defendant persona in a complete judgment document. To facilitate the evaluation of our approach, we collect complete judgment documents of criminal cases from China Judgments Online and PKU Fabao,[2] and construct three datasets named as CJO-S, CJO-L, and PKU, respectively. We extract the defendant persona, fact description, charges, law articles, and prison term using regular expressions. CJO-S and CJO-L datasets contain 10 distinct charges and 15 distinct law articles, whereas the PKU dataset contains 20 distinct charges and 28 distinct law articles. The prison terms are divided into the same 11 nonoverlapping intervals as the RCAIL dataset.

For the four datasets, we randomly select 80% of all the cases for training process, the remaining cases are equally distributed to validation set (10%) and testing set (10%). We list detailed statistics of the four datasets in Table 1.

B. EXPERIMENTAL SETTINGS

We employ jieba[3] for Chinese word segmentation and set the maximum document length as 1,000 words. We train the word embeddings on all the judgment documents and set the embedding size as 200. The hidden state size of GRU is set as 100 for each direction in the Bi-GRU sequence encoders. We set the embedding sizes of case fact, law article and defendant persona as 100, 50 and 50.

For the training, we use Adam [28] as the optimizer, which works well in gradient-based optimization problems with

[2]http://www.pkulaw.cn/

[3]https://github.com/fxsjy/jieba

TABLE 2. LJP results on the RCAIL dataset.

Tasks	Charges				Law Articles				Prison Term			
Metrics	AUC	Prec.	Rec.	F1	AUC	Prec.	Rec.	F1	AUC	Prec.	Rec.	F1
TF-IDF+SVM	0.887	0.788	0.791	0.783	0.752	0.551	0.461	0.493	0.601	0.460	0.294	0.286
CNN	0.898	0.830	0.789	0.801	0.794	0.602	0.572	0.589	0.627	0.448	0.369	0.308
GRU	0.912	0.879	0.842	0.827	0.808	0.606	0.637	0.622	0.654	0.429	0.317	0.327
Bi-GRU	0.917	0.873	0.850	0.836	0.833	0.594	0.650	0.658	0.660	0.451	0.340	0.339
HAN	0.933	0.899	0.846	0.867	0.898	0.837	0.810	0.799	0.673	0.469	0.323	0.368
TOPJUDGE	0.945	0.933	0.857	0.893	0.905	0.835	0.812	0.810	0.681	0.467	0.368	0.375
MANN	**0.955**	**0.949**	**0.902**	**0.912**	**0.913**	**0.846**	**0.829**	**0.825**	**0.693**	**0.478**	**0.379**	**0.402**

TABLE 3. LJP results on the CJO-S dataset.

Tasks	Charges				Law Articles				Prison Term			
Metrics	AUC	Prec.	Rec.	F1	AUC	Prec.	Rec.	F1	AUC	Prec.	Rec.	F1
TF-IDF+SVM	0.878	0.764	0.743	0.755	0.729	0.503	0.476	0.474	0.571	0.391	0.254	0.242
CNN	0.894	0.811	0.796	0.787	0.786	0.572	0.582	0.575	0.610	0.423	0.292	0.298
GRU	0.902	0.869	0.855	0.810	0.783	0.556	0.629	0.569	0.629	0.412	0.301	0.309
Bi-GRU	0.909	0.873	0.832	0.816	0.804	0.561	0.657	0.616	0.639	0.441	0.314	0.318
HAN	0.920	0.883	0.826	0.837	0.891	0.811	0.796	0.782	0.653	0.463	0.331	0.325
TOPJUDGE	0.937	0.914	0.851	0.876	0.902	0.819	0.808	0.803	0.670	0.454	0.350	0.366
MANN	**0.949**	**0.945**	**0.907**	**0.901**	**0.924**	**0.890**	**0.851**	**0.842**	**0.705**	**0.502**	**0.391**	**0.412**

TABLE 4. LJP results on the CJO-L dataset.

Tasks	Charges				Law Articles				Prison Term			
Metrics	AUC	Prec.	Rec.	F1	AUC	Prec.	Rec.	F1	AUC	Prec.	Rec.	F1
TF-IDF+SVM	0.892	0.788	0.769	0.784	0.763	0.539	0.501	0.515	0.592	0.413	0.265	0.261
CNN	0.897	0.833	0.810	0.791	0.784	0.580	0.589	0.577	0.621	0.429	0.286	0.303
GRU	0.910	0.871	0.847	0.815	0.791	0.577	0.645	0.580	0.633	0.438	0.319	0.311
Bi-GRU	0.913	0.879	0.826	0.823	0.826	0.621	0.669	0.632	0.645	0.461	0.330	0.325
HAN	0.922	0.896	0.833	0.841	0.909	0.840	0.818	0.811	0.663	0.470	0.347	0.339
TOPJUDGE	0.949	0.924	0.873	0.895	0.917	0.852	0.827	0.829	0.685	0.466	0.367	0.379
MANN	**0.963**	**0.955**	**0.931**	**0.923**	**0.936**	**0.913**	**0.875**	**0.872**	**0.719**	**0.521**	**0.433**	**0.429**

sparse gradients and has a theoretically better convergence rate than many other stochastic optimization methods. The learning rate as 0.001. The learning rate, dropout rate, and batch size are set as 0.001, 0.5, and 32, respectively. The training process will be terminated if there is no performance improvement over the validation set for successive 10 epochs.

For the evaluation, we employ *macro Area Under Curve* [29] (AUC), *macro Precision* (Prec.), *macro Recall* (Rec.), and *macro F1-score* (F1) as metrics. The macro-level metrics are calculated by averaging the AUC, precision, recall, and F1-score of each category, which is a means of highlighting the model's performance on infrequent classes.

C. BASELINES

To evaluate the performance of the proposed MANN framework, we compare our approach with the following models:

- **TF-IDF+SVM**: an SVM text classifier with word-level TF-IDF features [14], [17].
- **CNN**: a CNN-based classification model with multiple filter widths [30].
- **GRU**: a two-layer GRU network as the sequence encoder [26].
- **Bi-GRU**: a two-layer Bi-GRU network as the sequence encoder.
- **HAN**: a Hierarchical Attention Network for Document Classification [11].
- **TOPJUDGE**: a topological multitask learning framework for LJP [23].

D. RESULTS AND DISCUSSION

Experimental results of the LJP task on test sets of the four datasets are shown in Table 2, 3, 4, 5, from which we can observe that the proposed MANN framework outperforms all the baseline models for the three subtasks.

We present the relative performance gain in terms of F1 against the closest baseline on the four datasets in Fig. 6. Compared with TOPJUDGE, which performs best among all the baselines, MANN enhances the F1 by 2.99%, 4.54%, and 11.72% on average for the subtasks of charge prediction, law article prediction, and prison term prediction, respectively. The advantage of MANN relative to TOPJUDGE lies in that, instead of formalizing dependencies among the LJP subtasks in a fixed framework, our approach uses loss weights to control the training of three predictors and guides the multi-channel attentive encoders to produce task-oriented case representations, which is more scalable and robust in real-world scenarios.

The results demonstrate that most models achieve better LJP performance on the RCAIL dataset than that on CJO-S,

TABLE 5. LJP results on the PKU dataset.

Tasks	Charges				Law Articles				Prison Term			
Metrics	AUC	Prec.	Rec.	F1	AUC	Prec.	Rec.	F1	AUC	Prec.	Rec.	F1
TF-IDF+SVM	0.872	0.772	0.739	0.752	0.743	0.520	0.487	0.485	0.587	0.411	0.271	0.259
CNN	0.879	0.809	0.778	0.763	0.772	0.568	0.563	0.549	0.603	0.393	0.281	0.289
GRU	0.891	0.836	0.823	0.785	0.779	0.541	0.617	0.553	0.624	0.406	0.288	0.303
Bi-GRU	0.897	0.849	0.810	0.795	0.805	0.552	0.644	0.597	0.638	0.420	0.309	0.315
HAN	0.912	0.851	0.811	0.816	0.866	0.793	0.762	0.747	0.641	0.451	0.325	0.319
TOPJUDGE	0.926	0.897	0.846	0.854	0.890	0.803	0.783	0.780	0.664	0.446	0.343	0.359
MANN	**0.941**	**0.938**	**0.910**	**0.887**	**0.918**	**0.862**	**0.834**	**0.829**	**0.698**	**0.489**	**0.380**	**0.409**

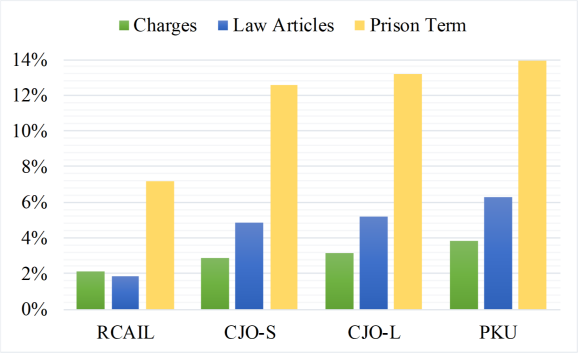

FIGURE 6. Relative performance gain (%) of F1 on the four datasets.

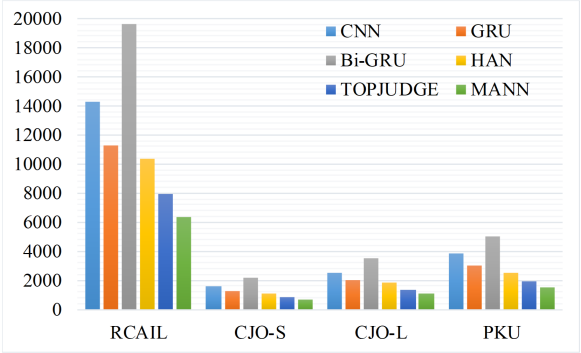

FIGURE 7. Training time (s) of neural network models on the four datasets.

since training data in RCAIL are more abundant. The only exception is our MANN framework, which has shown more prominent improvements in the law article and prison term prediction subtasks on the CJO-S dataset. This is because MANN can benefit from the defendant persona only contained in CJO-S and whereby attentively learns better representations of the involved specific circumstances, and ultimately makes a more credible prediction of law articles and prison terms. When it comes to the CJO-L dataset, it contains the same numbers of distinct charges, law articles, and prison term intervals as CJO-S, but the larger number of training data leads to the better performance than that on CJO-S for all the models. However, the larger training set of the PKU dataset has not improved the LJP performance as CJO-L does. This can be explained by the problem of data imbalance coming with the increased categories of charges and law articles.

The other baseline models, i.e., TF-IDF+SVM, CNN, GRU, and HAN, perform worse than MANN and TOPJUDGE, since they treat the three parts of LJP task as independent tasks, without utilizing the correlation among them. This indicates the significance of performing the integrated LJP task in a unified framework. That RNN-based models achieve better performance than CNN model shows the advantage of RNNs in processing sequential textual data. The HAN model stands out from the rest due to the hierarchical structure representation of judgment documents. Although TF-IDF+SVM attains a pretty high accuracy especially on the charge prediction, neural network models have better performance on macro metrics, which indicates that neural network is a better way to capture latent semantic features from different kinds of case descriptions.

TABLE 6. Ablation study on the CJO-S dataset.

Tasks	Charges		Law Articles		Prison Term	
Metrics	AUC	F1	AUC	F1	AUC	F1
MANN	**0.949**	**0.901**	**0.924**	**0.842**	**0.705**	**0.412**
w/o attention	0.927	0.857	0.895	0.792	0.663	0.353
w/o article	0.935	0.878	0.883	0.766	0.659	0.337
w/o persona	0.938	0.881	0.887	0.779	0.632	0.309

We also evaluate the training time of all the neural network models on the four datasets. As shown in Fig. 7, the attentive models (HAN, TOPJUDGE, MANN) are faster, since the introduction of attention mechanism efficiently promotes the convergence rate. The proposed MANN framework takes slightly less time to converge than TOPJUDGE while achieving higher prediction accuracy, which should be attributed to the multichannel attentive sequence encoder that learns better representations of case descriptions and the logical dependencies among three LJP subtasks.

E. ABLATION STUDY

Our approach is characterized by the incorporation of multichannel attention mechanism and hierarchical sequence encoders oriented to different parts of judgment documents. Thus, we design ablation test to evaluate the effectiveness of these modules.

As shown in Table 6, "w/o attention" is to remove the attention layers from the MANN framework, which means all the sentences and words are used to produce sequence embedding; in this case, the performance decreases on all LJP subtasks, which confirms again that the introduction of attention

mechanism has improved the semantic representation of judgment documents.

When taken off law article encoders as ''w/o article'', our approach conducts the LJP task based on only the case descriptions extracted from judgment documents, the performance degradation verifies the significance of employing law articles as input and legal basis to enhance the comprehension and interpretation of case facts.

Furthermore, ''w/o persona'' denotes removing persona encoders from the MANN framework. Here, we merge the defendant persona and case fact and feed them into the fact encoder together, whereas the LJP performance still encounters a significant decline especially on the law article and prison term prediction subtasks. This indicates that a specific encoder is essential for MANN to take full advantage of the defendant persona and leverage it to make a refined prediction that conforms to the involved specific circumstances.

From the above results, we may safely draw the conclusion that all of the modules in our approach can advance the model performance and a combination of them will achieve better results on the LJP task.

VI. CONCLUSION

In this paper, we propose MANN, a multichannel attentive neural network that can carry out the multiple parts of the LJP task in a unified framework. The incorporation of attention mechanism and hierarchical sequence encoders is adopted to learn better semantic representations and interactions among different parts of case descriptions. The experimental results on four real-world datasets of criminal cases in mainland China show that, our approach significantly outperforms all the baseline models and achieves state-of-the-art performance on the entire LJP task.

However, there is still a clear gap between the macro-level performance of prison term prediction and that of the other two subtasks, which shows that our approach suffers heavily from the imbalanced classes of prison terms. In addition, the criminal case with multiple defendants remains too complicated for our approach to deal with. We leave these challenges for further research.

REFERENCES

[1] B. Hachey and C. Grover, ''Extractive summarisation of legal texts,'' *Artif. Intell. Law*, vol. 14, no. 4, pp. 305–345, 2006.
[2] T. Gonçalves and P. Quaresma, ''Evaluating preprocessing techniques in a text classification problem,'' in *Proc. BCS*, São Leopoldo, Brazil, 2005, pp. 841–850.
[3] R. M. Palau and M. Moens, ''Argumentation mining: The detection, classification and structure of arguments in text,'' in *Proc. ICAIL*, Barcelona, Spain, 2009, pp. 98–107.
[4] C. Liu, C. Chang, and J. Ho, ''Case instance generation and refinement for case-based criminal summary judgments in Chinese,'' *J. Inf. Sci. Eng.*, vol. 20, no. 4, pp. 783–800, 2004.
[5] C. Liu and C. Hsieh, ''Exploring phrase-based classification of judicial documents for criminal charges in chinese,'' in *Proc. ISMIS*, Bari, Italy, 2006, pp. 681–690.
[6] D. M. Katz, M. J. Bommarito, II, and J. Blackman, ''A general approach for predicting the behavior of the Supreme Court of the United States,'' *PLoS ONE*, vol. 12, no. 4, 2017, Art. no. e0174698.
[7] W. Lin, T. Kuo, T. Chang, C. Yen, C. Chen, and S. Lin, ''Exploiting machine learning models for chinese legal documents labeling, case classification, and sentencing prediction,'' *IJCLCLP*, vol. 17, no. 4, pp. 49–68, 2012.
[8] B. Luo, Y. Feng, J. Xu, X. Zhang, and D. Zhao, ''Learning to predict charges for criminal cases with legal basis,'' in *Proc. EMNLP*, Copenhagen, Denmark, 2017, pp. 2727–2736.
[9] S. Long, C. Tu, Z. Liu, and M. Sun, ''Automatic judgment prediction via legal reading comprehension,'' 2018, *arXiv:1809.06537*. [Online]. Available: https://arxiv.org/abs/1809.06537
[10] H. Ye, X. Jiang, Z. Luo, and W. Chao, ''Interpretable charge predictions for criminal cases: Learning to generate court views from fact descriptions,'' in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, New Orleans, LA, USA, Jan. 2018, pp. 1854–1864.
[11] Z. Yang, D. Yang, C. Dyer, X. He, A. J. Smola, and E. H. Hovy, ''Hierarchical attention networks for document classification,'' in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, San Diego, CA, USA, Jun. 2016, pp. 1480–1489.
[12] D. Bahdanau, K. Cho, and Y. Bengio, ''Neural machine translation by jointly learning to align and translate,'' in *Proc. ICLR*, San Diego, CA, USA, 2015, pp. 1–15.
[13] Z. Yao, J. Yao, and W. Sun, ''Adaptive RISE control of hydraulic systems with multilayer neural-networks,'' *IEEE Trans. Ind. Electron.*, vol. 66, no. 11, pp. 8638–8647, Nov. 2019.
[14] N. Aletras, D. Tsarapatsanis, D. Preoţiuc-Pietro, and V. Lampos, ''Predicting judicial decisions of the European Court of Human Rights: A natural language processing perspective,'' *PeerJ Comput. Sci.*, vol. 2, p. e93, Oct. 2016.
[15] O. Şulea, M. Zampieri, M. Vela, and J. van Genabith, ''Predicting the law area and decisions of french supreme court cases,'' in *Proc. RANLP*, Varna, Bulgaria, 2017, pp. 716–722.
[16] O. Şulea, M. Zampieri, S. Malmasi, M. Vela, L. P. Dinu, and J. van Genabith, ''Exploring the Use of Text Classification in the Legal Domain,'' in *Proc. ASAIL*, London, U.K., 2017, pp. 1–5.
[17] G. Boella, L. D. Caro, and L. Humphreys, ''Using classification to support legal knowledge engineers in the eunomos legal document management system,'' in *Proc. 5th Int. Workshop Juris-Inform.*, 2011, pp. 1–12.
[18] Y. Liu and Y. Chen, ''A two-phase sentiment analysis approach for judgement prediction,'' *J. Inf. Sci.*, vol. 44, no. 5, pp. 594–607, 2018.
[19] C. Liu and T. Liao, ''Classifying criminal charges in chinese for web-based legal services,'' in *Proc. Asia–Pacific Web Conf.*, Shanghai, China, 2005, pp. 64–75.
[20] Y.-H. Liu, Y.-L. Chen, and W.-L. Ho, ''Predicting associated statutes for legal problems,'' *Inf. Process. Manage.*, vol. 51, no. 1, pp. 194–211, 2015.
[21] I. Chalkidis and I. Androutsopoulos, ''A deep learning approach to contract element extraction,'' in *Proc. JURIX*, Luxembourg City, Luxembourg, 2017, pp. 155–164.
[22] F. Wei, H. Qin, S. Ye, and H. Zhao, ''Empirical study of deep learning for text classification in legal document review,'' in *Proc. IEEE Int. Conf. Big Data*, Seattle, WA, USA, Dec. 2018, pp. 3317–3320.
[23] H. Zhong, Z. Guo, C. Tu, C. Xiao, Z. Liu, and M. Sun, ''Legal judgment prediction via topological learning,'' in *Proc. EMNLP*, Brussels, Belgium, 2018, pp. 3540–3549.
[24] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, ''Distributed representations of words and phrases and their compositionality,'' in *Proc. Adv. Neural Inf. Process. Syst.*, Lake Tahoe, NV, USA, 2013, pp. 3111–3119.
[25] S. Hochreiter and J. Schmidhuber, ''Long short-term memory,'' *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[26] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, ''Learning phrase representations using RNN encoder-decoder for statistical machine translation,'' in *Proc. EMNLP*, Doha, Qatar, 2014, pp. 1724–1734.
[27] C. Xiao, H. Zhong, Z. Guo, C. Tu, Z. Liu, M. Sun, Y. Feng, X. Han, Z. Hu, H. Wang, and J. Xu, ''CAIL2018: A large-scale legal dataset for judgment prediction,'' 2018, *arXiv:1807.02478*. [Online]. Available: https://arxiv.org/abs/1807.02478
[28] D. P. Kingma and J. Ba, ''Adam: A method for stochastic optimization,'' in *Proc. ICLR*, San Diego, CA, USA, 2015, pp. 1–41.
[29] C. Buckley and E. M. Voorhees, ''Retrieval evaluation with incomplete information,'' in *Proc. 27th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Sheffield, U.K., 2004, pp. 25–32.
[30] Y. Kim, ''Convolutional neural networks for sentence classification,'' in *Proc. EMNLP*, 2014, pp. 1746–1751.

SHANG LI received the B.S. and M.S. degrees in computer science from the Harbin Institute of Technology (HIT), Harbin, China, in 2012 and 2014, respectively, where he is currently pursuing the Ph.D. degree in computer science. His research interests include legal intelligence, data mining, and social network analysis.

HONGLI ZHANG received the B.S. degree in computer science from Sichuan University, Chengdu, China, in 1994, and the Ph.D. degree in computer science from the Harbin Institute of Technology (HIT), Harbin, China, in 1999. In 2012, she was a Visiting Scholar with North Carolina State University, Raleigh, NC, USA. She is currently a Professor with the School of Computer Science and Technology, HIT and the Vice Director of the National Computer Information Content Security Key Laboratory. Her research interests include network and information security, network measurement and modeling, and parallel processing.

LIN YE received the Ph.D. degree from the Harbin Institute of Technology, Harbin, China, in 2011. From January 2016 to January 2017, he was a Visiting Scholar with the Department of Computer and Information Sciences, Temple University, USA. His current research interests include network security, peer-to-peer networks, network measurement, and cloud computing.

XIAODING GUO received the B.S. degree in computer science from Xidian University, Xi'an, China, in 2015. She is currently pursuing the Ph.D. degree in computer science with the Harbin Institute of Technology. Her research interests include big data, data mining, and artificial intelligence.

BINXING FANG received the Ph.D. degree from the Harbin Institute of Technology, Harbin, China, in 1989. He is currently a member of the Chinese Academy of Engineering. His research interests include cyberspace security, big data, and cloud computing.