Information Processing and Management 57 (2020) 102305

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

An end-to-end joint model for evidence information extraction from court record document

Donghong Ji[a], Peng Tao[a], Hao Fei[a], Yafeng Ren[*,b]

[a] School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China
[b] Laboratory of Language and Artificial Intelligence, Guangdong University of Foreign Studies, Guangzhou 510420, China

ARTICLE INFO

Keywords:
Natural language processing
Information extraction
Court record document
Neural networks
Joint model

ABSTRACT

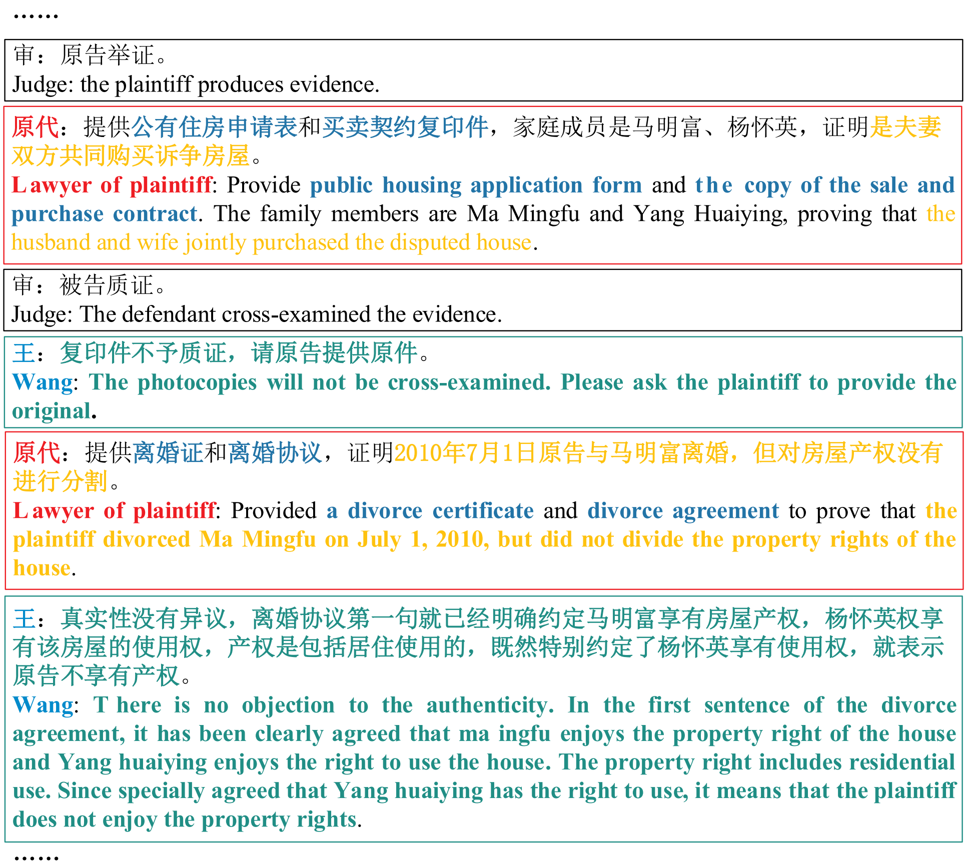
Information extraction is one of the important tasks in the field of Natural Language Processing (NLP). Most of the existing methods focus on general texts and little attention is paid to information extraction in specialized domains such as legal texts. This paper explores the task of information extraction in the legal field, which aims to extract evidence information from court record documents (CRDs). In the general domain, entities and relations are mostly words and phrases, indicating that they do not span multiple sentences. In contrast, evidence information in CRDs may span multiple sentences, while existing models cannot handle this situation. To address this issue, we first add a classification task in addition to the extraction task. We then formulate the two tasks as a multi-task learning problem and present a novel end-to-end model to jointly address the two tasks. The joint model adopts a shared encoder followed by separate decoders for the two tasks. The experimental results on the dataset show the effectiveness of the proposed model, which can obtain 72.36% F1 score, outperforming previous methods and strong baselines by a large margin.

1. Introduction

Legal text mining aims to automatically analyze the texts in the legal domain by employing various NLP techniques (Giacalone, Cusatelli, Romano, Buondonno, & Santarcangelo, 2018; Qazi & Wong, 2019; Srinivasa & Thilagam, 2019), and is becoming one heated research topic, such as legal text generation (Alschner & Skougarevskiy, 2017; Kanapala, Pal, & Pamula, 2019; Merchant & Pande, 2018; Polsley, Jhunjhunwala, & Huang, 2016), legal judgment prediction (Branting et al., 2019; Chalkidis, Androutsopoulos, & Aletras, 2019; Li, Zhang, Ye, Guo, & Fang, 2019; Zhong et al., 2018), legal text classification (Chalkidis, Fergadiotis, Malakasiotis, Aletras, & Androutsopoulos, 2019; Elnaggar, Gebendorfer, Glaser, & Matthes, 2018; Li, Wang, & Ma, 2019) and legal query understanding (Kumar & Politi, 2019; Shankar & Buddarapu, 2018; 2019a), etc. In recent years, information extraction in the legal domain has received increasing attention from researchers. For example, Buey, Garrido, Bobed, and Ilarri (2016) use a special type of ontology, and reference to relevant extraction mechanisms to guide the detection of specific data in Spanish legal documents. Garcia-Constantino et al. (2017) create a CLIEL (Commercial Law Information Extraction based on Layout) environment using hybrid techniques to detect several types of information in commercial law documentations. Nguyen, Nguyen, Tojo, Satoh, and Shimazu (2018) use a Bi-LSTM-CRF (Bidirectional Long Short-term Memory Conditional Random Field) model to detect necessary and effect parts in Japan legal documents. More recently, Leitner, Rehm, and Moreno-Schneider (2019) treat it as a named entity recognition (NER) task, and use multiple NER approaches to detect semantic concepts in German court decisions.

[*] Corresponding author.
E-mail address: renyafeng@whu.edu.cn (Y. Ren).

https://doi.org/10.1016/j.ipm.2020.102305
Received 7 February 2020; Received in revised form 14 May 2020; Accepted 15 May 2020
Available online 29 May 2020



……

审：原告举证。
Judge: the plaintiff produces evidence.

原代：提供公有住房申请表和买卖契约复印件，家庭成员是马明富、杨怀英，证明是夫妻双方共同购买诉争房屋。
Lawyer of plaintiff: Provide **public housing application form** and **the copy of the sale and purchase contract**. The family members are Ma Mingfu and Yang Huaiying, proving that the husband and wife jointly purchased the disputed house.

审：被告质证。
Judge: The defendant cross-examined the evidence.

王：复印件不予质证，请原告提供原件。
Wang: **The photocopies will not be cross-examined. Please ask the plaintiff to provide the original.**

原代：提供离婚证和离婚协议，证明2010年7月1日原告与马明富离婚，但对房屋产权没有进行分割。
Lawyer of plaintiff: Provided **a divorce certificate** and **divorce agreement** to prove that **the plaintiff divorced Ma Mingfu on July 1, 2010, but did not divide the property rights of the house**.

王：真实性没有异议，离婚协议第一句就已经明确约定马明富享有房屋产权，杨怀英权享有该房屋的使用权，产权是包括居住使用的，既然特别约定了杨怀英享有使用权，就表示原告不享有产权。
Wang: **T here is no objection to the authenticity. In the first sentence of the divorce agreement, it has been clearly agreed that ma ingfu enjoys the property right of the house and Yang huaiying enjoys the right to use the house. The property right includes residential use. Since specially agreed that Yang huaiying has the right to use, it means that the plaintiff does not enjoy the property rights**.

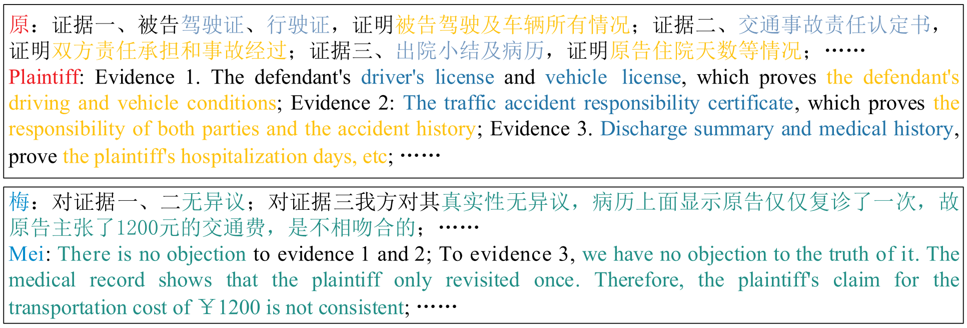
……

Fig. 1. The main part of a CRD. The speaker before the colon refers to the judge or party. The paragraphs in the black, red and green boxes indicate their type, and they are other, *evidence production* and *evidence cross-examination* paragraph, respectively. The bold and colored sections are key evidence information to be extracted. Red words represent *evidence provider*, blue words represent *evidence name* and orange words represent *evidence content*, cyan words represent *cross-examination party* and green words represent *cross-examination opinion*. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Different from the existing work, the research objective of this paper is to explore a new problem in legal texts: evidence information extraction. This paper aims to extract evidence information from count record documents (CRDs) by designing a novel neural network model. Specifically, we extract five types of evidence information from a CRD, which consists of *evidence provider (EP), evidence name (EN), evidence content (EC), cross-examination party (CEP)* and *cross-examination opinion (CEO)*. Our research can help experts and professionals in the legal field avoid a lot of manual labor and better understand legal texts. Meanwhile, our work can facilitate the application of machine learning algorithms, especially neural network models, in the legal field.

As illustrated in Fig. 1, a CRD mainly contains judge, plaintiff, defendant, lawyers and their utterances. Specifically, the evidence provider "原代" (lawyer of plaintiff), provides the evidence names, "住房申请表" (public housing application form) and "买卖契约复印件" (the copy of the sale and purchase contract), to support the content including the family members and their home purchase. Furthermore, the cross-examining party "王" (Wang, the defendant), gives his opinion that he does not accept the copy of the form. Therefore, the goal of evidence information extraction is to detect key evidence information (coloured parts in Fig. 1) provided by the parties that can be used to help the judge make a decision.

Some preliminary work has been proposed for information extraction in legal texts (Dutta, Das, & Chakraborty, 2020; Goularte, Nassar, Fileto, & Saggion, 2019). However, they fails to achieve satisfactory results by applying the existing methods directly to the task. First, unlike general texts, some sentences that contain key information are longer in CRDs and vary in length depending on the identity of the parties. As shown in Fig. 1, the judge's statements are concise, while the parties' statements are longer and more complex. In addition, evidence information in CRDs may span multiple sentences. For Wang's statements shown in Fig. 1, the cross-examination opinion of the defendant can be very lengthy. However, most of the existing methods focus on the sentence-level extraction and do not handle our task well. Second, existing methods generally treat evidence information extraction as a NER task, ignoring the latent category information behind CRD data that is useful for the extraction task. Taking the evidence paragraph and the cross-examination paragraph of Fig. 2 as example, the paragraph above is the *evidence production* paragraph, and the below is the *evidence cross-examination* paragraph. We can find that *evidence production* paragraph contains three types of evidence information (*evidence provider, evidence name* and *evidence content*), and *evidence cross-examination* paragraph contains another two types of evidence information (*cross-examination party* and *cross-examination opinion*). Meanwhile, if a paragraph contains *evidence provider,*



原：证据一、被告驾驶证、行驶证，证明被告驾驶及车辆所有情况；证据二、交通事故责任认定书，证明双方责任承担和事故经过；证据三、出院小结及病历，证明原告住院天数等情况；……

Plaintiff: Evidence 1. The defendant's driver's license and vehicle license, which proves the defendant's driving and vehicle conditions; Evidence 2: The traffic accident responsibility certificate, which proves the responsibility of both parties and the accident history; Evidence 3. Discharge summary and medical history, prove the plaintiff's hospitalization days, etc; ……

梅：对证据一、二无异议；对证据三我方对其真实性无异议，病历上面显示原告仅仅复诊了一次，故原告主张了1200元的交通费，是不相吻合的；……

Mei: There is no objection to evidence 1 and 2; To evidence 3, we have no objection to the truth of it. The medical record shows that the plaintiff only revisited once. Therefore, the plaintiff's claim for the transportation cost of ￥1200 is not consistent; ……

Fig. 2. A more detailed example of *evidence production* paragraph and *evidence cross-examination* paragraph. Red word represent *evidence provider*, blue words represent *evidence name* and orange words represent *evidence content*, cyan words represent *cross-examination party* and green words represent *cross-examination opinion*. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

evidence name and *evidence content*, it is impossible to contain other two types of evidence information, and vice versa. So the paragraph's category can potentially provide useful information for the final extraction.

To address the above issues, this paper proposes a novel method for evidence information extraction in legal texts. We first divide evidence information from CRDs into several predefined evidence categories, and then divide the paragraphs into several predefined paragraph categories based on evidence categories. Based on the predefined categories, we propose an end-to-end joint model (named JBLACN), which first adopts a shared encoder followed by separate decoders for paragraph classification and evidence extraction. The intermediate information in the paragraph classification module can be also used in the evidence extraction module. In such architecture, the intermediate representation of predefined paragraph categories can be effectively leveraged into the evidence extraction module for facilitating the final extraction. Furthermore, long short-term memory network (LSTM) can handle the sequences of any length without the problem of gradients vanishing and explosion (Bengio, Simard, Frasconi et al., 1994; Hochreiter & Schmidhuber, 1997), so we use a bidirectional LSTM (Bi-LSTM) as a shared encoder in JBLACN. Meanwhile, not all input items contribute equally to the representation of a sequence, so we use the attention mechanism to automatically select the most important items. Finally, we tackle our task on the paragraph level so that we can extract evidence information that spans multiple sentences.

The experimental dataset is manually collected from real-world courts in China from 2013 to 2019. The experimental results on the dataset show that our model achieves 72.36% F1 score, significantly outperforming baseline systems, and demonstrating the effectiveness of the proposed model. Our codes and datasets are released at https://github.com/Justprogramer/JBLACN.

In summary, the main contributions of the paper are as follows:

- We explore a new task of information extraction, which aims to extract evidence information from court record documents.
- We formalize evidence information extraction task as an integration of paragraph classification and sequence labeling problem, and propose an end-to-end joint model for the task.
- Results on CRDs show that our model achieves the current best performance, outperforming previous methods and strong baselines by a large margin.

2. Related work

2.1. Legal text mining

In recent years, text mining in the legal domain has attracted increasing attention from the NLP community (Pérez-Rodríguez, Pérez-Pérez, Fdez-Riverola, & Lourenço, 2019). For example, Polsley et al. (2016) provide a user interface for automated text summarization of a legal document, and use a number of word-frequency based pipeline modules, with additional domain-specific knowledge. Bajwa, Karim, Naeem, and ul Amin (2017) investigate a semi-supervised classification algorithm to extract catchphrases from software license agreements based on parts-of-speech (POS) tagging, morphological analysis and syntactic parsing. Zhong et al. (2018) propose a multi-task learning framework that divides legal judgement prediction into multiple subtasks and formalizes the explicit dependencies on these subtasks into a directed acyclic graph. Shankar and Buddarapu (2019b) regard legal query reformulation as a monolingual neural machine translation problem and apply an encoder-decoder framework combined with attention learning to improve the quality of search engine results. More recently, Chalkidis, Androutsopoulos et al. (2019) verify three subtasks in judgment prediction of English legal texts using neural networks.

Among legal text mining, legal information extraction has gradually attracted the attention of researchers (Eirini & Grigorios, 2018). For example, Buey et al. (2016) use a special type of ontology to detect specific data on Spanish legal documents. Garcia-Constantino et al. (2017) employ hybrid technologies to detect several types of information on commercial law documentations. Nguyen et al. (2018) treat the task as a sequence labeling problem, and use several Bi-LSTM-CRF models to detect necessary and effect parts of Japan legal documents. Furthermore, Leitner et al. (2019) formulate it as a NER task, and use several NER approaches to detect semantic concepts in German court decisions. Barrière and Fouret (2019) design a two-step learning approach to improve the performance of a state-of-the-art NER model that leverages contextual information by automatically generating a context dictionary of entities. However, most of the existing methods focus on sentence-level extraction, and fail to give satisfactory results by directly applying these models in our task.

2.2. Named entity recognition

Our work is also closely related to named entity recognition (NER), which is one fundamental task in information extraction (Fei, Ren, & Ji, 2019; Qian, Deng, Ye, Ma, & Yuan, 2019). Existing NER approaches generally regard it as a sequence labeling problem (Fei, Ren, & Ji, 2020; Ren, Fei, & Ren, 2018; Zhang & Yang, 2018; Zheng et al., 2017). Currently, representative models include probabilistic graph models such as Conditional Random Fields (CRF) (Ratinov & Roth, 2009) and deep neural networks such as Recurrent Neural Network (RNN) and Convolutional Neural Networks (CNN) (Collobert et al., 2011; Hammerton, 2003; dos Santos & Guimarães, 2015). It should be noted that recent studies largely utilize the LSTM-CRF architecture. For example, Huang, Xu, and Yu (2015) use the LSTM-CRF framework with hand-crafted spelling features for the task. Ma and Hovy (2016) use a character CNN to represent spelling features. Besides, some studies devote to improve the performance by using external resources and more carefully-designed networks. Specifically, Peters, Ammar, Bhagavatula, and Power (2017) add pre-trained contextualized embeddings from a bidirectional language model for improving the performance. Peters et al. (2018) learn linear combinations of internal hidden states stacked in a deep bidirectional language model, to utilize both high-level states which capture context-dependent aspects and low-level states in modeling syntax. Zhang and Yang (2018) use a lattice structured LSTM-CRF for NER disambiguation.

More recently, Xia et al. (2019) use a detector and a classifier to address the problem of non-overlapping or nested NER, rather than treating it as a sequential labeling task and annotating entities consecutively. Cui and Zhang (2019) investigate a hierarchically-refined Bi-LSTM-LAN model for sequence labeling and demonstrate that it can effectively address the problem of label bias. Different from the above methods, we propose an end-to-end joint model for evidence information extraction in legal texts. Note that our model is inspired by these models, but differs in motivation and structure.

3. Task modeling

Different from the previous methods (Cheng, Li, Ren, Lou, & Gao, 2019; Dai et al., 2019; Luan, Ostendorf, & Hajishirzi, 2017), which directly model evidence information extraction as a sequential labeling task, in this study we regard it as a combination task of intermediate paragraph classification and final sequence labeling. Specifically, the intermediate paragraph classification task is defined as: for a document D containing m paragraphs $\{P_1, P_2, \ldots, P_m\}$ and a candidate classification label set $T = \{t_1, t_2, \ldots, t_{|T|}\}$, the goal is to predict each paragraph label $\hat{t_i}$ in D:

$$\hat{t_i} = f(P_i; \theta_c), \tag{1}$$

where $\hat{t_i} \in T$, $f(\cdot)$ and θ_c denote the classification module and its parameters, respectively.

In this paper, we divide evidence information extracted from paragraphs into five categories: *evidence provider (EP), evidence name (EN), evidence content (EC), cross-examination party (CEP)* and *cross-examination opinion (CEO)*. To facilitate the extraction, we classify paragraphs into two categories, namely, *evidence production* and *evidence cross-examination*. From the *evidence production* paragraph, the three evidence categories (*EP, EN* and *EC*) can be extracted. For the *evidence cross-examination* paragraph, the latter evidence categories (*CEP* and *CEO*) can be extracted.

Similarly, sequence labeling task is defined as: for a paragraph containing n words $P_j = \{w_{j1}, w_{j2}, \ldots, w_{jn}\}$ and a candidate label set $L = \{l_1, l_2, \cdots, l_{|L|}\}$, the goal is to predict each word label $\hat{l_i}$ in P_j:

$$\hat{l_i} = g(w_{ji}; \theta_s), \tag{2}$$

where $\hat{l_i} \in L$, $g(\cdot)$ and θ_s denote sequence labeling module and its parameters, respectively.

Once the results of sequence labeling module are obtained, evidence information can be extracted based on these labels. Note that the results of the classification module have no effect on the extraction task, we need the results of sequence labeling module, but taking both results into account in training can improve the performance of the extraction. That is because the categories of paragraphs can help sequence labeling module capture more features from CRDs. The experimental results also validate our idea.

4. Method

As described in Section 3, each paragraph is classified as an *evidence production* (or a *evidence cross-examination*) type, indicating that evidence (or cross-examination) information is contained in the paragraph, and vice versa. Note that the classification and sequence labeling are performed simultaneously. The architecture of the proposed model is shown in Fig. 3, which consists of an embedding layer, an encoding layer, a paragraph extraction module for sequence labeling and a paragraph classification module.

4.1. Embedding layer

Given a document containing m paragraphs $D = \{P_1, P_2, P_3, \cdots, P_m\}$, the paragraph P_j containing n words is a word sequence P_j=$\{w_{j1}, w_{j2}, w_{j3}, \cdots, w_{jn}\}$. Each word can be represented by its embedding $\boldsymbol{x}_{ji} = \boldsymbol{e}^w(w_{ji})$, where $\boldsymbol{e}^w(\cdot)$ denotes a word embedding lookup

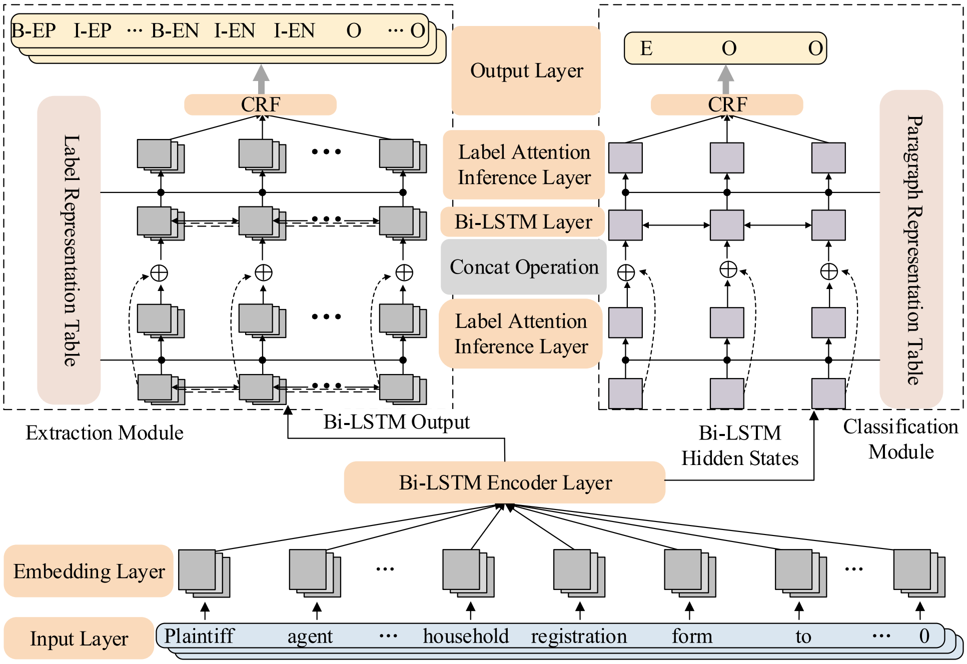

B-EP I-EP ··· B-EN I-EN I-EN O ··· O
CRF
Label Representation Table
Extraction Module
Bi-LSTM Output
Output Layer
Label Attention Inference Layer
Bi-LSTM Layer
Concat Operation
Label Attention Inference Layer
E O O
CRF
Paragraph Representation Table
Bi-LSTM Hidden States
Classification Module
Bi-LSTM Encoder Layer
Embedding Layer
Input Layer
Plaintiff agent ··· household registration form to ··· 0

Fig. 3. The architecture of the proposed model. The shared part is a LSTM layer. The left is the extraction module and the right is the classification module. Each side consists of 2 Bi-LSTM-LAN layers. The input of the figure is a document, which consists of three paragraphs.

table (Li et al., 2018). So the paragraph P_j is represented as a sequence of vectors $\boldsymbol{X}_j = \{\boldsymbol{x}_{j1}, \boldsymbol{x}_{j2}, \boldsymbol{x}_{j3}, ..., \boldsymbol{x}_{jn}\}$, and the document D is represented as a sequence of vectors $\boldsymbol{X} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \boldsymbol{X}_3, ..., \boldsymbol{X}_m\}$.

4.2. Encoding layer

In this paper, we use a bidirectional LSTM (Bi-LSTM) to learn the representation from the word sequence layer. RNN (Recurrent Neural Network) is a type of neural networks that operate on sequential data, and is widely used for processing sequential information. Theoretically, RNN can make use of the information in arbitrarily long sequences, but in practice, the standard RNN suffers from the problem of vanishing gradients (Bengio et al., 1994). This makes it difficult to model long-distance correlation in a sequence. LSTM solves this problem by using memory gates and forgetting gates to better capture long-range dependencies (Lample, Ballesteros, Subramanian, Kawakami, & Dyer, 2016). For a sequence of vectors $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, ..., \boldsymbol{x}_n\}$, LSTM can generate the corresponding vector representations $\{\boldsymbol{h}_1, \boldsymbol{h}_2, \boldsymbol{h}_3, ..., \boldsymbol{h}_n\}$. Basically, a LSTM represents each time step with an input, a memory and an output gate, denoted as i_t, f_t and o_t, respectively.

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_{ii}\boldsymbol{x}_t + b_{ii} + \boldsymbol{W}_{hi}\boldsymbol{h}_{(t-1)} + b_{hi}), \tag{3}$$

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}_{if}\boldsymbol{x}_t + b_{if} + \boldsymbol{W}_{hf}\boldsymbol{h}_{(t-1)} + b_{hf}), \tag{4}$$

$$\boldsymbol{g}_t = \tanh(\boldsymbol{W}_{ig}x_t + b_{ig} + \boldsymbol{W}_{hg}\boldsymbol{h}_{(t-1)} + b_{hg}), \tag{5}$$

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_{io}\boldsymbol{x}_t + b_{io} + \boldsymbol{W}_{ho}\boldsymbol{h}_{(t-1)} + b_{ho}), \tag{6}$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t\boldsymbol{c}_{(t-1)} + \boldsymbol{i}_t\boldsymbol{g}_t, \tag{7}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \tanh(\boldsymbol{c}_t), \tag{8}$$

where $\sigma(\cdot)$ is the element-wise sigmoid function.

Bi-LSTM consists of a forward LSTM and a backward LSTM, each of which can generate a sequence. The forward LSTM reads the word from left to right, so its output is represented as $\{\overrightarrow{\boldsymbol{h_1}}, \overrightarrow{\boldsymbol{h_2}}, \overrightarrow{\boldsymbol{h_3}}, ..., \overrightarrow{\boldsymbol{h_n}}\}$. In contrast, the backward LSTM reads the word from right to left, so its output is represented as $\{\overleftarrow{\boldsymbol{h_1}}, \overleftarrow{\boldsymbol{h_2}}, \overleftarrow{\boldsymbol{h_3}}, ..., \overleftarrow{\boldsymbol{h_n}}\}$. Finally, we use these two vectors to concatenate a new representation, so the ith word $\boldsymbol{x}_{ji}$ in the jth paragraph P_j is represented as $\boldsymbol{h}_{ji} = [\overrightarrow{\boldsymbol{h_{ji}}}; \overleftarrow{\boldsymbol{h_{ji}}}]$ and the jth paragraph's output in this layer is represented as $\boldsymbol{H}_j = \{\boldsymbol{h}_{j1}, \boldsymbol{h}_{j2}, \boldsymbol{h}_{j3}, ..., \boldsymbol{h}_{jn}\}$.

Since these paragraphs are classified simultaneously, not only these words, but also these paragraphs need to be encoded. We concatenate the last output of the forward LSTM $[\overrightarrow{\boldsymbol{h_{jn}}}]$ and the last output of the backward LSTM $[\overleftarrow{\boldsymbol{h_{j1}}}]$, to generate a representation of the jth paragraph $\boldsymbol{p}_j = [\overrightarrow{\boldsymbol{h_{jn}}}; \overleftarrow{\boldsymbol{h_{j1}}}]$. Finally, a document are represented as $\boldsymbol{H} = \{\boldsymbol{H}_1, \boldsymbol{H}_2, \boldsymbol{H}_2, \cdots, \boldsymbol{H}_m\}$, and the paragraphs of the document are represented as $\boldsymbol{P} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3, \cdots, \boldsymbol{p}_m\}$. Then, the vectors in $\boldsymbol{H}$ are used as the input of the extraction module and $\boldsymbol{P}$ is sent to the classification module.

4.3. Extraction module and classification module

We model the extraction task and the classification task as a sequence labeling task, so we can use the same module for these two tasks. We use a label attention network (LAN) which performs the attention over label embeddings for deriving a marginal label distribution (Cui & Zhang, 2019), which is in turn used to calculate a weighted sum of label embeddings. As shown in Fig. 3, this module consists of stacked attentive Bi-LSTM layers, each of which is composed of a Bi-LSTM layer and a label attention inference layer, and takes a sequence of vectors as input and yields a sequence of hidden state vectors together with a sequence of label distributions. Finally, the packed label vector with input word vectors is used together as the hidden state vector for the current layer. For sequence labeling, the input is a paragraph and the output is the label distribution of each word in the final layer.

4.3.1. Label representation

For a set of candidate output labels $L = \{l_1, l_2, l_3, \ldots, l_{|L|}\}$, each label can be represented as a vector by a label embedding lookup table:

$$\boldsymbol{x}_k^l = \boldsymbol{e}^l(l_k), \tag{9}$$

where $\boldsymbol{e}^l$ denotes a label embedding lookup table. We randomly initialize the label embeddings and fine-tune it during training.

4.3.2. Bi-LSTM layer

As described in Section 4.2, Bi-LSTM is used to compute $\boldsymbol{H} \in \mathbb{R}^{m\times n\times d_h}$ and $\boldsymbol{P} \in \mathbb{R}^{m\times d_h}$, where m, n and d_h denote the number of paragraphs, the word sequence length and the hidden size of Bi-LSTM (the same dimension as the label embedding), respectively.

4.3.3. Label attention inference layer

In this layer, an attention matrix $\boldsymbol{\alpha}$ is produced by the attention mechanism, which represents the potential label distribution for each word in a paragraph. For the jth paragraph, we define $\boldsymbol{Q} = \boldsymbol{H}_j$ and $\boldsymbol{K} = \boldsymbol{V} = \boldsymbol{x}^l.\boldsymbol{x}^l \in \mathbb{R}^{l\times d_h}$ is the label set representation and l denotes the size of labels.

$$\boldsymbol{H}_j^l = Attention(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \boldsymbol{\alpha}\boldsymbol{V}, \tag{10}$$

$$\boldsymbol{\alpha} = softmax(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_h}}). \tag{11}$$

Besides, we use a multi-head attention mechanism for capturing multiple possible label distributions in parallel. The specific implementations are as follows:

$$\begin{aligned}\boldsymbol{H}_j^l &= MultiHead(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V})\\ &= [head_1; \ldots; head_n]\boldsymbol{W}^O,\end{aligned} \tag{12}$$

$$head_i = Attention(\boldsymbol{Q}\boldsymbol{W}_i^Q, \boldsymbol{K}\boldsymbol{W}_i^K, \boldsymbol{V}\boldsymbol{W}_i^V), \tag{13}$$

where [;] denotes a concatenation operation, $\boldsymbol{W}_i^Q \in \mathbb{R}^{d_h\times\frac{d_h}{k}}$, $\boldsymbol{W}_i^K \in \mathbb{R}^{d_h\times\frac{d_h}{k}}$ and $\boldsymbol{W}_i^V \in \mathbb{R}^{d_h\times\frac{d_h}{k}}$ are parameters which need to be learned during training and k is the number of parallel heads.

The concatenation of Bi-LSTM hidden states and the attention is represented as $\boldsymbol{H}_j = [\boldsymbol{H}_j^l; \boldsymbol{H}_j]$, which is the final representation of the jth paragraph for each Bi-LSTM-LAN layer, if any, and will be then fed to a subsequent Bi-LSTM-LAN layer as input.

4.4. Inference layer

In the last layer, the extraction (or classification) module predicts the label of each word (or paragraph) using CRF. For the jth input paragraph sequence $\boldsymbol{X}_j = \{\boldsymbol{x}_{j1}, \boldsymbol{x}_{j2}, \boldsymbol{x}_{j3}, \ldots, \boldsymbol{x}_{jn}\}$, $\boldsymbol{H}_j$ is an output of the final Bi-LSTM-LAN layer, which is size of $n \times l$, where l is the number of distinct tags, and $\boldsymbol{H}_j^{p,q}$ corresponds to the score of the qth tag of the pth word in a paragraph. For a sequence of predictions $y = (y_1, y_2, y_3, \ldots, y_n)$, its score is defined as:

$$s(\boldsymbol{X}_j, y) = \sum_{i=0}^{n} \boldsymbol{A}_{y_i, y_{i+1}} + \sum_{i=1}^{n} \boldsymbol{H}_j^{i,y_i}, \tag{14}$$

where $\boldsymbol{A}$ is a matrix of transition scores, $\boldsymbol{A}_{i,j}$ represents the score of a transition from the tag i to j. y_0 and y_n are the *start* and *end* tags of a sentence, respectively, so $\boldsymbol{A}$ is a square matrix of size $k + 2$.

All possible tag sequences pass through a softmax layer, yielding a probability for the sequence y:

$$p(y|\boldsymbol{X}_j) = \frac{e^{s(\boldsymbol{X}_j, y)}}{\sum_{\tilde{y} \in Y_{X_j}} e^{s(\boldsymbol{X}_j, \tilde{y})}}, \tag{15}$$

where Y_{X_j} represents all possible sequence labels given an input sequence $\boldsymbol{X}_j$. Finally, the output sequence that obtains the maximum score is given by:

$$y^{\star} = \underset{\tilde{y} \in Y_{X_j}}{\arg\max}\, s(\boldsymbol{X}_j, \tilde{y}). \tag{16}$$

4.5. Training

Given an input sequence $\boldsymbol{X}_j$, the negative log-probability of the correct label sequence y is defined as:

$$\begin{aligned} Loss(\boldsymbol{X}_j;\ \theta) &= -\log(p(y|\boldsymbol{X}_j;\ \theta)) \\ &= -s(\boldsymbol{X}_j, y;\ \theta) + \log\Bigg(\sum_{\tilde{y} \in Y_{X_j}} e^{s(\boldsymbol{X}_j, \tilde{y};\ \theta)}\Bigg), \end{aligned} \tag{17}$$

where θ denotes the model parameters.

Because the model contains two modules: the extraction module and the classification module, each of them has a loss score. Here, $Loss_s(\cdot)$ and $Loss_c(\cdot)$ represent the objective of the extraction module and the classification module, respectively, obtained by Eq. (17). The final objective is computed as:

$$Loss_{all}(\boldsymbol{X}_j;\ \theta) = \beta \cdot Loss_s(\boldsymbol{X}_j;\ \theta_s) + (1-\beta) \cdot Loss_c(\boldsymbol{X}_j;\ \theta_c), \tag{18}$$

where θ_s and θ_c denote the model parameters and β denotes the weight parameter of two modules. $Loss_{all}(\cdot)$ is minimized during training.

5. Experimental settings

5.1. Dataset

The experimental dataset is annotated by experts in the legal field and contains 1128 CRDs from several provinces and cities in China from 2013 to 2019. As previously described, our goal is to extract five types of evidence information, so that each word in the sentence can be labeled by the *BIO* encoding format, which is shown in Table 1. We randomly divide these documents into a training set, a development set, and a test set with a ratio of 8:1:1. Then we extract the main parts of these documents with rules and divide them into paragraphs, the detailed statistics are shown in Table 2.

5.2. Baseline systems

To show the effectiveness of the proposed model, we compare our model with the following baseline systems:

- *Bi-LSTM-CRF*, is proposed by Lample et al. (2016), which contains an embedding layer, an encoding layer and an inference layer.
- *Bi-LSTM-LAN*, is proposed by Cui and Zhang (2019), with or without CRF.
- *Pipeline*, consists of a paragraph classification module TextCNN proposed by Kim (2014), and two extraction modules implemented by Bi-LSTM-CRF or Bi-LSTM-LAN-CRF.
- *LM-Bi-LSTM-JNT*, is proposed by Ye and Ling (2018), which uses a joint architecture of hybrid semi-Markov CRF (HSCRF) and CRF.

For pipeline methods, all paragraphs of a document are first marked by using TextCNN. The evidence paragraphs are then fed into the evidence model, and the cross-examination paragraphs are fed into the cross-examination model. Note that these two models have the same effect as Bi-LSTM-CRF or Bi-LSTM-LAN-CRF, except that the elements responsible for the extraction are different.

Table 1
The used labels in a sentence.

Information category	Labels
Evidence provider	B/I-EP
Evidence name	B/I-EN
Evidence content	B/I-EC
Cross-examination party	B/I-CEP
Cross-examination opinion	B/I-CEO
Other	O

Table 2
Statistics of the dataset.

Category	Training	Dev.	Test	Total
Document	902	113	113	1,128
Paragraph categories				
Evidence paragraph	2342	269	305	2916
Cross-examination paragraph	2840	341	349	3530
Other paragraph	23,850	3442	2626	29,918
Total	29,032	4,052	3,280	36,364
Information categories				
Evidence provider	2169	256	295	2720
Evidence name	7431	945	766	9142
Evidence content	4606	642	390	5638
Cross-examination party	2840	341	349	3530
Cross-examination opinion	4221	525	430	5176
Total	21,267	2709	2230	26,206

5.3. Parameter settings

We build JBLACN based on NCRF++ (Yang & Zhang, 2018), and parameters of JBLACN are shown in Table 3. We use 300-dimensional Chinese word embeddings (Li et al., 2018) in the embedding layer. In the Bi-LSTM-LAN sub-layer, we use 300-dimensional hidden states for both Bi-LSTM and label embedding, and we use 2 Bi-LSTM-LAN sub-layers because experimental results show that more layers do not improve the performance. We optimize the model with SGD and initial learning rate is set to 0.02 with a decay rate 0.05. The dropout rate is set to 0.5 and the multi-head number is set to 5. Each batch is a document because it preserves the sequential relationship between paragraphs. To take advantage of two modules, a weight parameter β is introduced. we conduct development experiments to find the best β, which is shown in Fig. 4. We can find that the model achieves the best performance when it is set to 0.8.

5.4. Evaluation metrics

We use widely used evaluation metrics: precision, recall and $F1$ score. $F1$ score is the harmonic mean of precision and recall. Specifically, the annotated fragment is saved with its absolute location number (start and end position) and category. For example, *EN[3,10]* means this fragment is *evidence name*, consisting of the third to the tenth words of the sentence. The start position index, end position index and category are compared with the gold standard of the annotation, and if they are exactly the same, the recognition result is considered as correct. Besides, we also use BLEU and ROUGE metrics to evaluate different models for partially-correct extraction.

6. Experimental results

In this section, we empirically compare Bi-LSTM-CRF, pipeline method, Bi-LSTM-LAN (with or without CRF) and our JBLACN. The results of different methods are shown in Table 4. We have the following several observations.

First, we can see that Bi-LSTM-CRF achieves 60.42% $F1$ score, and Bi-LSTM-LAN without CRF gives 63.69% $F1$ score. By integrating CRF, Bi-LSTM-LAN-CRF gives 68.47% $F1$ score. Compared with Bi-LSTM-CRF, Bi-LSTM-LAN and Bi-LSTM-LAN-CRF both give better performance, increasing by 3.27% and 8.05%, respectively, which shows that Bi-LSTM-LAN and Bi-LSTM-LAN-CRF are more effective than Bi-LSTM-CRF. Besides, Bi-LSTM-LAN-CRF significantly outperforms Bi-LSTM-LAN, giving an improvement of 4.78% in $F1$ score. The main reason is that CRF can successfully capture the dependencies among the output labels for identifying key phrases and better contribute to the full model's predictions.

Second, in pipeline methods, Bi-LSTM-CRF(+TextCNN) achieves 66.19% $F1$ score, which is higher (5.77%) than that of Bi-LSTM-CRF. The obvious improvement proves that classifying paragraphs first is helpful for the subsequent extraction task. Compared with Bi-LSTM-CRF(+TextCNN), Bi-LSTM-LAN-CRF(+TextCNN) gives 1.29% improvement in $F1$ score, showing that the effect of the classification is greater than the effect of hierarchical label attention mechanism. However, $F1$ score of Bi-LSTM-LAN-CRF

Table 3
Parameters of JBLACN.

Parameter	Value	Parameter	Value
Word embedding	300	Bidirectional	True
Label embedding	300	Bi-LSTM-LAN layers	2
Hidden size	300	Dropout	0.5
Learning rate	0.02	Multi-head	5
Decay rate	0.05	Module weight	0.8

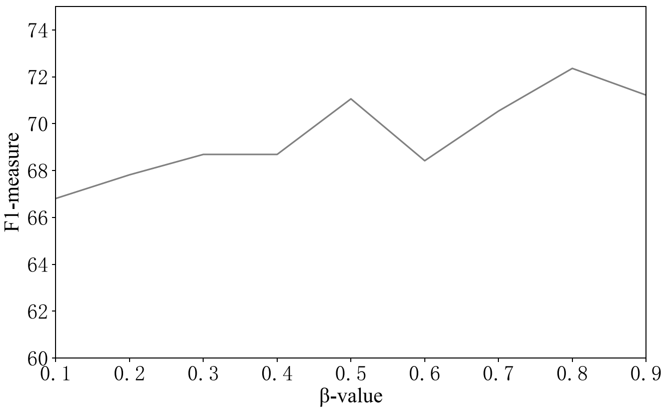


Fig. 4. Results of different β values.

Table 4
Results of different models. The results with marker * demonstrates that the p value is below 10^{-3} using t-test compared with the best system Bi-LSTM-LAN-CRF.

Method	Precision (%)	Recall (%)	*F*1 (%)
Bi-LSTM-CRF	67.88	53.80	60.42
Bi-LSTM-LAN	63.71	63.68	63.69
Bi-LSTM-LAN-CRF	70.06	66.95	68.47
Pipeline			
Bi-LSTM-CRF(+ TextCNN)	68.44	64.08	66.19
Bi-LSTM-LAN-CRF(+ TextCNN)	68.16	66.82	67.48
LM-Bi-LSTM-JNT	68.71	54.53	60.81
JBLACN	**73.64**	**71.12**	**72.36***

(+ TextCNN) is lower than that of Bi-LSTM-LAN-CRF, which indicates that error propagation affects the final performance in pipeline methods. Compared with Bi-LSTM-CRF(+ TextCNN), Bi-LSTM-LAN gives a 2.5% reduction in *F*1 score but Bi-LSTM-LAN-CRF achieves 2.28% improvement, which shows that both CRF and classification have an important effect on the extraction task. The above analysis shows the effectiveness of CRF and the paragraph categories for the final extraction.

Third, we also compare our model with a joint model LM-Bi-LSTM-JNT, which is a hybrid semi-Markov conditional random filed architecture for neural sequence labeling (Ye & Ling, 2018). We can find that LM-Bi-LSTM-JNT achieves 60.81% *F*1 score, while our proposed model JBLACN gives 72.36% *F*1 score, increasing by nearly 11.55% than the former. For the former, the LM-Bi-LSTM-JNT model uses a hybrid CRF-HSCRF architecture, which only uses sequence tag information for joint training. The model cannot use paragraph category information, so the effect is similar to the Bi-LSTM-CRF model. This shows that it is quite different between our task and sequence labeling in the general domain, and the JBLACN model is better for evidence information extraction in this paper. Note that JBLACN increases by nearly 6.17% over Bi-LSTM-CRF(+ TextCNN) and 4.88% over Bi-LSTM-LAN-CRF(+ TextCNN), which prove that error propagation exists in pipeline methods again. Compared with Bi-LSTM-LAN-CRF, JBLACN increases by 3.89% in *F*1 score. These improvements indicate that the paragraph classification module does make contribution to the evidence extraction module when they are trained simultaneously. The above analysis shows that JBLACN can achieve significant improvements for evidence information extraction, which also indicates the paragraph classification and the paragraph extraction are mutually reinforced.

Finally, to show the ability of the proposed model, we further conducts experiments to evaluate our model JBLACN and the best baseline system Bi-LSTM-LAN-CRF based on metrics BLEU and ROUGE, which is shown in Table 5. Different from *F*1 score that calculates the results of accurate extraction, BLEU and ROUGE can judge the ability of the model for partially-correct extraction. Based on Table 5, we can find that our model JBLACN outperforms the state-of-the-art Bi-LSTM-LAN-CRF model by a large margin based on BLEU and ROUGE. The above analysis shows the proposed model is more effective than previous methods in evidence information extraction from court record documents.

Table 5
Results of two models on BLEU and ROUGE (abbreviated to R).

Model	BLEU (%)	R-1 (%)	R-2 (%)	R-L (%)
Bi-LSTM-LAN-CRF	58.31	66.65	63.97	66.55
JBLACN	61.47	70.90	68.08	70.82

6.1. Results of different categories

To better analyze the extraction results, we show the performance of each category from our model JBLACN. The results are shown in Table 6, the last row denotes the average value of all categories. Note that *evidence provider* achieves the highest $F1$ score (87.54%), while *cross-examination opinion* achieves the lowest $F1$ score (57.77%). We can find that the evidence paragraph has more significant effect than the cross-examination paragraph. The main reason is not only that the statement of the evidence paragraph has obvious characteristics while the statement in the cross-examination paragraph is not obvious, but also related to the contextual characteristics of CRDs. We will give more detailed analysis based on some examples in Section 6.4.

6.2. Ablation study

We analyze the effect of various components of JBLACN by replacing or removing components. The results are presented in Table 7.

We first analyze the results of different embeddings. We explore the state-of-the-art language models BERT (Devlin, Chang, Lee, & Toutanova, 2019) and ELMO (Peters et al., 2018). When replacing embedding layer with BERT (Devlin et al., 2019) or ELMO (Peters et al., 2018), the performance of the model degrades drastically, the main reason is that we use BERT and ELMO as static embedding lookup table.[1] While our embedding layer with fine-tuning gives better performance than static BERT and ELMO. Furthermore, when removing different components from JBLACN one by one, such as CRF, LAN and classification module, the performance of the model also degrades drastically. Especially, when the LAN module is removed, the performance decreases even more, which indicates that the LAN module makes a significant contribution to word embeddings and paragraph representations. With the LAN module's contribution, the CRF module is more capable to fully capture this representation. Besides, removing the classification module has a greater effect than the removal of the CRF module, which indicates that reasonable weight parameters between the extraction module and the classification module make the classification module more useful than the CRF module.

When removing multiple components from JBLACN at the same time, the performance of the model is even worse. Interestingly, we find that removing only the LAN module has a greater impact than removing both the CRF and LAN modules. We think that this is due to the lack of LAN module's contribution to the representation, which makes the CRF module fail to capture the clues of some words, leading to the boundary prediction errors. Besides, when the CRF and classification modules or LAN and classification modules or CRF, LAN and classification modules are removed, our JBLACN will become the Bi-LSTM-LAN model or Bi-LSTM-CRF or Bi-LSTM, respectively. It is reasonable that these effects have declined. Based on the above analysis, the ablation study shows that the components of JBLACN are reasonable and introducing the paragraph classification can lead to the improved performance for evidence information extraction.

6.3. Attention visualization

We show the visualization of attention weights to verify the effectiveness of the attention mechanism in Fig. 5. Because we use the label attention mechanism, the horizontal axis represents the input sequence, and the vertical axis represents the output labels. We use a brighter color to represent higher weights of the corresponding words in a paragraph. As shown in the heatmap, "车辆维修清单" (vehicle maintenance checklist), "发票" (invoice) and "刷卡支付回执单" (credit card payment receipt) are clearly *evidence name*, so the B/I-EN labels achieve higher weights than other labels for these words. What's more, "原告发生交通事故事故后产生的维修费用为XXXX元" (the maintenance cost incurred by the plaintiff after the traffic accident is XXXX yuan) is *evidence content*, which is the detailed description for the evidences just listed. These words more focus on the B/I-EC labels. The above analysis shows that the label attention mechanism is capable of learning higher weights for more valid words.

6.4. Error analysis

Although the proposed model JBLACN performs better than other baselines, there are still some categories that are not extracted well. To understand the limitations of our model for further improvement, we analyze some cases and summarize several types of errors.

First, whether the current paragraph is cross-examination or not depends on the context. For example, "无异议" (No objection) is not a *cross-examination opinion* in the context shown in Fig. 6(a), but in the context shown in Fig. 6(b), it is a *cross-examination opinion* expressed on "驾驶证和行驶证" (driver's license and vehicle license). Although this situation is considered in our model, there is still some influences for the extraction. This is the main reason why $F1$ value of *cross-examination party* is 74.63% in Table 6 which is much lower than *evidence provider*.

Second, for *cross-examination opinion* shown in Table 6, $F1$ score is only 57.77%. The main reason is that it is difficult to determine *cross-examination opinion*'s actual boundary. As shown in Fig. 6(c), *cross-examination opinion* sometimes begins with a word in the middle of a sentence and ends after a few sentences. If the beginning or end position is wrong, the extraction is usually a wrong result. To prove our conjecture, we conduct experiments by extending the beginning position of *cross-examination opinion* to the first word of

[1] we tried to use fine-tuned BERT in our embedding layer, but it will run out of our graphics memory. The main reason is that some paragraphs are too long (more than 512 characters), if we do not freeze these parameters, it is hard to use them as our embedding layer for comparison.

Table 6
Results of different categories.

Category	Precision (%)	Recall (%)	*F*1 (%)
Evidence provider	89.40	85.76	87.54
Evidence name	76.52	77.02	76.77
Evidence content	67.37	65.64	66.49
Cross-examination party	76.90	72.49	74.63
Cross-examination opinion	60.58	55.21	57.77
Avg	74.15	71.22	72.64

Table 7
Results of ablation study.

Ablation method	Precision (%)	Recall (%)	*F*1 (%)
JBLACN	**73.64**	**71.12**	**72.36**
Replacing embedding layers			
+ BERT	67.94	68.79	68.36
+ ELMO	66.87	67.80	67.34
Removing components			
− CRF	69.58	69.92	69.75
− LAN	62.98	63.04	63.01
− Classification	70.06	66.95	68.47
− CRF and LAN	66.62	64.99	65.80
− CRF and classification	63.71	63.68	63.69
− LAN and classification	67.88	53.80	60.42
− CRF, LAN and classification	52.05	28.17	36.55

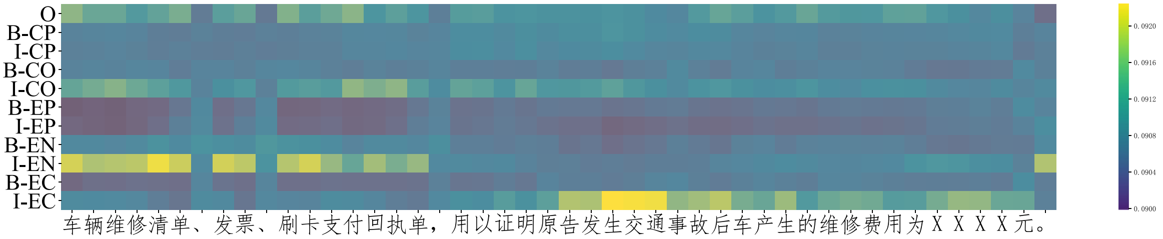


Fig. 5. The visualization of attention weights.

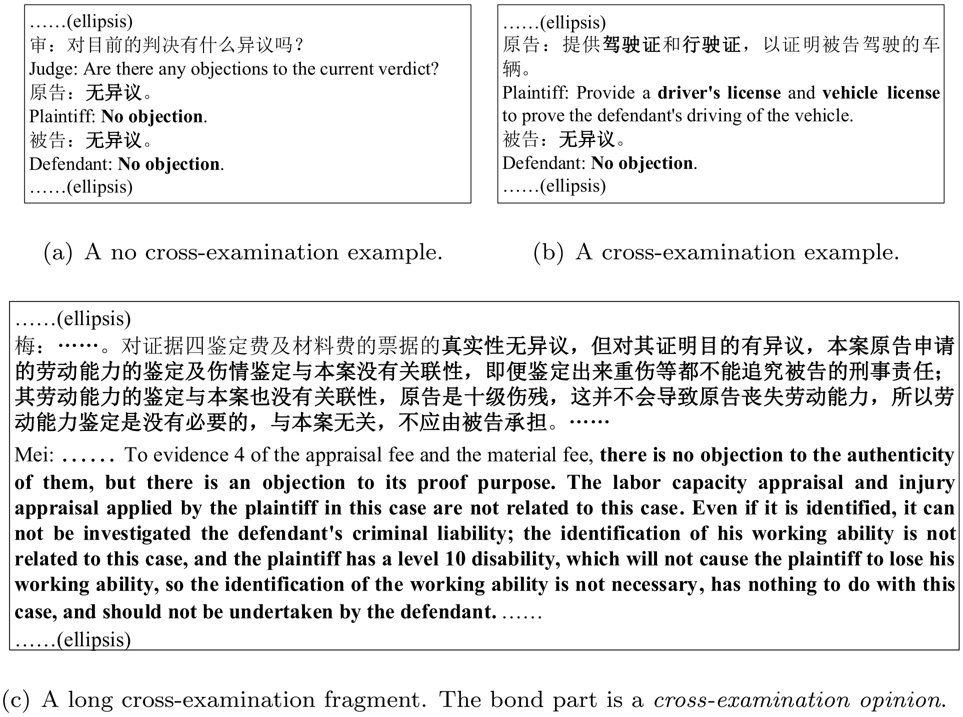
the current sentence. The new results of different categories are shown in Table 8. Compared to the former results, the performance is improved by 4.42% (*F*1 score) in *cross-examination opinion* category, although it decreases slightly in other categories. The new experimental results confirm our conjecture.

Third, for long-sentence paragraphs, it is difficult to extract all information correctly. Compared to *evidence name*, the lengths of *evidence content* and *cross-examination opinion* vary more extremely. For example, *cross-examination opinion* in Fig. 6(c) has 137 words, while Fig. 6(b) has only 3 words. As long as one of these words is labeled wrongly, it is a false extraction. How to deal with this extreme situation is one of the issues that need to be addressed in future work. This is one of the possible reasons why *F*1 values of *cross-examination party* and *evidence content* are low.

7. Conclusion

In this paper, we investigate an end-to-end joint model to extract evidence information from court record documents by regarding it as a combination task of intermediate paragraph classification and final sequence labeling. The experimental results on Chinese CRDs show the effectiveness of the proposed method, outperforming pervious methods and strong baseline systems by a large margin. The proposed model can be applied for better analyzing and understanding legal texts, avoiding a lot of manual labor by experts and professionals in the legal field.

As described about CRDs in Section 1, each *evidence name* has its *evidence content* and *cross-examination opinion*. So the task of evidence information extraction can be formalized as a quintuple: {*evidence provider, evidence name, evidence content, cross-examination party, cross-examination opinion*}, we will explore this in future work.



……(ellipsis)
审：对目前的判决有什么异议吗？
Judge: Are there any objections to the current verdict?
原告：**无异议**。
Plaintiff: **No objection**.
被告：**无异议**。
Defendant: **No objection**.
……(ellipsis)

(a) A no cross-examination example.

……(ellipsis)
原告：提供**驾驶证**和**行驶证**，以证明被告驾驶的车辆。
Plaintiff: Provide a **driver's license** and **vehicle license** to prove the defendant's driving of the vehicle.
被告：**无异议**。
Defendant: **No objection**.
……(ellipsis)

(b) A cross-examination example.

……(ellipsis)
梅：……。对证据四鉴定费及材料费的票据的**真实性无异议，但对其证明目的有异议，本案原告申请的劳动能力的鉴定及伤情鉴定与本案没有关联性，即便鉴定出来重伤等都不能追究被告的刑事责任；其劳动能力的鉴定与本案也没有关联性，原告是十级伤残，这并不会导致原告丧失劳动能力，所以劳动能力鉴定是没有必要的，与本案无关，不应由被告承担。**……
Mei: …… To evidence 4 of the appraisal fee and the material fee, **there is no objection to the authenticity of them, but there is an objection to its proof purpose. The labor capacity appraisal and injury appraisal applied by the plaintiff in this case are not related to this case. Even if it is identified, it can not be investigated the defendant's criminal liability; the identification of his working ability is not related to this case, and the plaintiff has a level 10 disability, which will not cause the plaintiff to lose his working ability, so the identification of the working ability is not necessary, has nothing to do with this case, and should not be undertaken by the defendant.** ……
……(ellipsis)

(c) A long cross-examination fragment. The bond part is a *cross-examination opinion.*

Fig. 6. Paragraphs examples for error analysis.

Table 8
New results of different categories.

Category	Precision (%)	Recall (%)	*F*1 (%)
Evidence provider	90.14	86.78	88.43
Evidence name	76.03	77.42	76.71
Evidence content	66.67	64.62	65.62
Cross-examination party	71.52	67.62	69.51
Cross-examination opinion	62.75	61.64	62.19
Avg	73.42	71.62	72.49

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Nos. 61702121 and 61772378), the Research Foundation of Ministry of Education of China (No. 18JZD015), the Key Project of State Language Commission of China (No. ZDI135-112), Guangdong Basic and Applied Basic Research Foundation of China (No. 2020A151501705) and the Science and Technology Project of Guangzhou (No. 201704030002).

References

Alschner, W., & Skougarevskiy, D. (2017). *Towards an automated production of legal texts using recurrent neural networks. Proceedings of the 16th edition of the international conference on artificial intelligence and law*229–232.

Bajwa, I. S., Karim, F., Naeem, M. A., & ul Amin, R. (2017). A semi supervised approach for catchphrase classification in legal text documents. *Journal of Computers, 12*(5), 451–461.

Barrière, V., & Fouret, A. (2019). *May I check again? – A simple but efficient way to generate and use contextual dictionaries for named entity recognition. application to french legal texts. Proceedings of the 22nd nordic conference on computational linguistics*327–332.

Bengio, Y., Simard, P., Frasconi, P., et al. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks, 5*(2), 157–166.

Branting, K., Weiss, B., Brown, B., Pfeifer, C., Chakraborty, A., Ferro, L., … Yeh, A. S. (2019). *Semi-supervised methods for explainable legal prediction. Proceedings of the seventeenth international conference on artificial intelligence and law*22–31.

Buey, M. G., Garrido, A. L., Bobed, C., & Ilarri, S. (2016). *The AIS project: Boosting information extraction from legal documents by using ontologies. Proceedings of the 8th* *international conference on agents and artificial intelligence*438–445.

Chalkidis, I., Androutsopoulos, I., & Aletras, N. (2019). *Neural legal judgment prediction in english. Proceedings of the 57th conference of the association for computational linguistics*4317–4323.

Chalkidis, I., Fergadiotis, E., Malakasiotis, P., Aletras, N., & Androutsopoulos, I. (2019). *Extreme multi-label legal text classification: A case study in EU legislation. Proceedings of the natural legal language processing workshop*78–87.

Cheng, M., Li, L., Ren, Y., Lou, Y., & Gao, J. (2019). A hybrid method to extract clinical information from chinese electronic medical records. *IEEE Access, 7*, 70624–70633.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research, 12*, 2493–2537.

Cui, L., & Zhang, Y. (2019). *Hierarchically-refined label attention network for sequence labeling. Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing*4106–4119.

Dai, D., Xiao, X., Lyu, Y., Dou, S., She, Q., & Wang, H. (2019). *Joint extraction of entities and overlapping relations using position-attentive sequence labeling. Proceedings of the thirty-third AAAI conference on artificial intelligence*6300–6308.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *Bert: Pre-training of deep bidirectional transformers for language understanding. Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics*4171–4186.

Dutta, S., Das, D., & Chakraborty, T. (2020). Changing views: Persuasion modeling and argument extraction from online discussions. *Information Processing & Management, 57*(2), 102085.

Eirini, P., & Grigorios, T. (2018). Local word vectors guiding keyphrase extraction. *Information Processing & Management, 54*(6), 888–902.

Elnaggar, A., Gebendorfer, C., Glaser, I., & Matthes, F. (2018). *Multi-task deep learning for legal document translation, summarization and multi-label classification. Proceedings of the 2018 artificial intelligence and cloud computing conference*9–15.

Fei, H., Ren, Y., & Ji, D. (2019). *Recognizing nested named entity in biomedical texts: A neural network model with multi-task learning. Proceedings of the 2019 IEEE international conference on bioinformatics and biomedicine*376–381.

Fei, H., Ren, Y., & Ji, D. (2020). Dispatched attention with multi-task learning for nested mention recognition. *Information Sciences, 513*, 241–251.

Garcia-Constantino, M., Atkinson, K., Bollegala, D., Chapman, K., Coenen, F., Roberts, C., & Robson, K. (2017). *CLIEL: Context-based information extraction from commercial law documents. Proceedings of the 16th international conference on artificial intelligence and law*79–87.

Giacalone, M., Cusatelli, C., Romano, A., Buondonno, A., & Santarcangelo, V. (2018). Big data and forensics: An innovative approach for a predictable jurisprudence. *Information Sciences, 426*, 160–170.

Goularte, F. B., Nassar, S. M., Fileto, R., & Saggion, H. (2019). A text summarization method based on fuzzy rules and applicable to automated assessment. *Expert Systems with Applications, 115*, 264–275.

Hammerton, J. (2003). *Named entity recognition with long short-term memory. Proceedings of the seventh conference on natural language learning*172–175.

Hochreiter, S., & Schmidhuber, J. (1997). *LSTM can solve hard long time lag problems. Proceedings of the annual conference on neural information processing systems*473–479.

Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. *CoRR* abs/1508.01991

Kanapala, A., Pal, S., & Pamula, R. (2019). Text summarization from legal documents: A survey. *Artificial Intelligence Review, 51*(3), 371–402.

Kim, Y. (2014). *Convolutional neural networks for sentence classification. Proceedings of the 2014 conference on empirical methods in natural language processing*1746–1751.

Kumar, S., & Politi, R. (2019). *Understanding user query intent and target terms in legal domain. Proceedings of the 24th international conference on applications of natural language to information systems*41–53.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). *Neural architectures for named entity recognition. Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics*260–270.

Leitner, E., Rehm, G., & Moreno-Schneider, J. (2019). *Fine-grained named entity recognition in legal documents. Proceedings of international conference on semantic systems*272–287.

Li, G., Wang, Z., & Ma, Y. (2019). Combining domain knowledge extraction with graph long short-term memory for learning classification of chinese legal documents. *IEEE Access, 7*, 139616–139627.

Li, S., Zhang, H., Ye, L., Guo, X., & Fang, B. (2019). Mann: A multichannel attentive neural network for legal judgment prediction. *IEEE Access, 7*, 151144–151155.

Li, S., Zhao, Z., Hu, R., Li, W., Liu, T., & Du, X. (2018). *Analogical reasoning on chinese morphological and semantic relations. Proceedings of the 56th annual meeting of the association for computational linguistics*138–143.

Luan, Y., Ostendorf, M., & Hajishirzi, H. (2017). *Scientific information extraction with semi-supervised neural tagging. Proceedings of the 2017 conference on empirical methods in natural language processing*2641–2651.

Ma, X., & Hovy, E. (2016). *End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. Proceedings of the 54th annual meeting of the association for computational linguistics*1064–1074.

Merchant, K., & Pande, Y. (2018). *NLP based latent semantic analysis for legal text summarization. Proceedings of the 2018 international conference on advances in computing, communications and informatics*1803–1807.

Nguyen, T.-S., Nguyen, L.-M., Tojo, S., Satoh, K., & Shimazu, A. (2018). Recurrent neural network-based models for recognizing requisite and effectuation parts in legal texts. *Artificial Intelligence and Law, 26*(2), 169–199.

Pérez-Rodríguez, G., Pérez-Pérez, M., Fdez-Riverola, F., & Lourenço, A. (2019). Online visibility of software-related web sites: The case of biomedical text mining tools. *Information Processing & Management, 56*(3), 565–583.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). *Deep contextualized word representations. Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics*2227–2237.

Peters, M. E., Ammar, W., Bhagavatula, C., & Power, R. (2017). *Semi-supervised sequence tagging with bidirectional language models. Proceedings of the 55th annual meeting of the association for computational linguistics*1756–1765.

Polsley, S., Jhunjhunwala, P., & Huang, R. (2016). *Casesummarizer: A system for automated summarization of legal texts. Proceedings of the 26th international conference on computational linguistics*258–262.

Qazi, N., & Wong, B. W. (2019). An interactive human centered data science approach towards crime pattern analysis. *Information Processing & Management, 56*(6), 102066.

Qian, Y., Deng, X., Ye, Q., Ma, B., & Yuan, H. (2019). On detecting business event from the headlines and leads of massive online news articles. *Information Processing & Management, 56*(6), 102086.

Ratinov, L., & Roth, D. (2009). *Design challenges and misconceptions in named entity recognition. Proceedings of the thirteenth conference on computational natural language learning*147–155.

Ren, Y., Fei, H., & Ren, H. (2018). *Neural networks for bacterial named entity recognition. Proceedings of the 2018 IEEE international conference on bioinformatics and biomedicine*2797–2799.

dos Santos, C. N., & Guimarães, V. (2015). *Boosting named entity recognition with neural character embeddings. Proceedings of the fifth named entity workshop*25–33.

Shankar, A., & Buddarapu, V. N. (2018). *Deep ensemble learning for legal query understanding. Proceedings of the 2018 workshops co-located with 27th ACM international conference on information and knowledge management*1–10.

Shankar, A., & Buddarapu, V. N. (Buddarapu, 2019a). *Legal query reformulation using deep learning. Proceedings of the third workshop on automated semantic analysis of information in legal texts co-located with the 17th international conference on artificial intelligence and law*1–10.

Shankar, A., & Buddarapu, V. N. (Buddarapu, 2019b). *Neural attention learning for legal query reformulation. Proceedings of the seventeenth international conference on artificial intelligence and law*272–273.

Srinivasa, K., & Thilagam, P. S. (2019). Crime base: Towards building a knowledge base for crime entities and their relationships from online news papers. *Information*

Processing & Management, 56(6), 102059.

Xia, C., Zhang, C., Yang, T., Li, Y., Du, N., Wu, X., ... Yu, P. S. (2019). *Multi-grained named entity recognition. Proceedings of the 57th conference of the association for computational linguistics*1430–1440.

Yang, J., & Zhang, Y. (2018). *NCRF++: An open-source neural sequence labeling toolkit. Proceedings of the 56th annual meeting of the association for computational linguistics*74–79.

Ye, Z., & Ling, Z.-H. (2018). *Hybrid semi-Markov CRF for neural sequence labeling. Proceedings of the 56th annual meeting of the association for computational linguistics*235–240.

Zhang, Y., & Yang, J. (2018). *Chinese NER using lattice LSTM. Proceedings of the 56th annual meeting of the association for computational linguistics*1554–1564.

Zheng, S., Wang, F., Bao, H., Hao, Y., Zhou, P., & Xu, B. (2017). *Joint extraction of entities and relations based on a novel tagging scheme. Proceedings of the 55th annual meeting of the association for computational linguistics*1227–1236.

Zhong, H., Guo, Z., Tu, C., Xiao, C., Liu, Z., & Sun, M. (2018). *Legal judgment prediction via topological learning. Proceedings of the 2018 conference on empirical methods in natural language processing*3540–3549.