Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

A full-process intelligent trial system for smart court*

Bin WEI†§1, Kun KUANG†§2, Changlong SUN†§2,3, Jun FENG†§4, Yating ZHANG3, Xinli ZHU5, Jianghong ZHOU2, Yinsheng ZHAI5, Fei WU†‡2

[1]*Guanghua Law School, Zhejiang University, Hangzhou 310008, China*

[2]*College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China*

[3]*Alibaba Group, Hangzhou 310099, China*

[4]*State Grid Zhejiang Electric Power Co., Ltd., Hangzhou 310007, China*

[5]*Zhejiang Higher People's Court, Hangzhou 310012, China*

†E-mail: srsysj@zju.edu.cn; kunkuang@zju.edu.cn; changlong.scl@taobao.com; JuneFeng.81@gmail.com; wufei@zju.edu.cn

Received Jan. 24, 2021; Revision accepted Aug. 3, 2021; Crosschecked Jan. 26, 2022

Abstract: In constructing a smart court, to provide intelligent assistance for achieving more efficient, fair, and explainable trial proceedings, we propose a full-process intelligent trial system (FITS). In the proposed FITS, we introduce essential tasks for constructing a smart court, including information extraction, evidence classification, question generation, dialogue summarization, judgment prediction, and judgment document generation. Specifically, the preliminary work involves extracting elements from legal texts to assist the judge in identifying the gist of the case efficiently. With the extracted attributes, we can justify each piece of evidence's validity by establishing its consistency across all evidence. During the trial process, we design an automatic questioning robot to assist the judge in presiding over the trial. It consists of a finite state machine representing procedural questioning and a deep learning model for generating factual questions by encoding the context of utterance in a court debate. Furthermore, FITS summarizes the controversy focuses that arise from a court debate in real time, constructed under a multi-task learning framework, and generates a summarized trial transcript in the dialogue inspectional summarization (DIS) module. To support the judge in making a decision, we adopt first-order logic to express legal knowledge and embed it in deep neural networks (DNNs) to predict judgments. Finally, we propose an attentional and counterfactual natural language generation (AC-NLG) to generate the court's judgment.

Key words: Intelligent trial system; Smart court; Evidence analysis; Dialogue summarization; Focus of controversy; Automatic questioning; Judgment prediction

https://doi.org/10.1631/FITEE.2100041 **CLC number:** TP391

1 Introduction

During the pandemic of COVID-19, online trials based on the intelligent trial system have become ubiquitous. The smart court relies on Internet courts to turn offline litigation activities into online activities. Online trials reduce the flow of personnel and keep trials in working order. The smart court has successfully implemented full-service online processing and built a comprehensive, multi-functional, and intensive online litigation platform, which has alleviated judicial urgency issues. The Supreme

§ These authors contributed equally to this work
‡ Corresponding author
* Project supported by the Key R&D Projects of the Ministry of Science and Technology of China (No. 2020YFC0832500), the National Key Research and Development Program of China (No. 2018AAA0101900), the National Social Science Foundation of China (No. 20&ZD047), the National Natural Science Foundation of China (Nos. 61625107 and 62006207), the Key R&D Project of Zhejiang Province, China (No. 2020C01060), and the Fundamental Research Funds for the Central Universities, China (Nos. LQ21F020020 and 2020XZA202)
ORCID: Bin WEI, https://orcid.org/0000-0002-6895-7007; Fei WU, https://orcid.org/0000-0003-2139-8807

People's Court promptly issued the "Notice on Strengthening and Standardizing Online Litigation during the COVID-19 Prevention and Control Period," which created a comprehensive deployment of online litigation for the courts to conduct proceedings with smart court. The smart court has formulated clear regulations for judicial tasks, such as online court hearings, electronic service, identity authentication, and material submission, and provided full judicial services and guarantees for online litigation promotion and regulation. According to the statistical data during the COVID-19 period (from February 3 to November 4, 2020), the people's courts at four levels filed 6.501 million online cases, 778 000 online court sessions, 3.23 million online mediations, and 18.15 million electronic services.

To make the smart court operate efficiently and improve trial efficiency in simple cases, Zhejiang Higher People's Court, Zhejiang University, and the Alibaba Group have jointly developed a full-process intelligent trial system (FITS), which provides strong technical support for constructing a smart court for the Zhejiang Provincial People's Court. FITS has played an essential role in financial lending and private lending cases, which moves the trial procedures of the court to the network platform, supports judicial trials in a highly informative manner, and assists judges in making judicial decisions. As shown in Fig. 1, the intelligent trial system implements the following judicial tasks: (1) extracting essential information from the legal text (indictment, lending contract, court debate transcript, etc.) to help the judge promptly grasp the key case information; (2) summarizing the controversy focuses from the court debate transcript recorded during the trial; (3) verifying the authenticity, legality, and relevance of the evidence; (4) recommending candidate questions to the judges to assist in the necessary trial procedures and discover facts related to the case; (5) retrieving the most similar cases from the historical data, and leveraging the knowledge of legal experts to predict case facts and help judges make judicial decisions; (6) generating a judgment document with complete structure, complete elements, and rigorous logic after confirming the facts of the case and applying laws and regulations.

Zhejiang University and the Alibaba Group have conducted much research on the above judicial tasks. Zhao et al. (2018) proposed a named entity recognition model based on the BiLSTM-CRF architecture, with two novel techniques of multi-task data selection and constrained decoding. Liu XJ et al. (2018) introduced a graph convolution based model to combine textual and visual information presented in visually rich documents (VRDs). Zhou et al. (2019) studied a novel research task of legal dispute judgment (LDJ) prediction for e-commerce transactions, which connects two isolated domains, e-commerce data mining and legal intelligence. Duan et al. (2019) introduced a delicately designed multi-role and multi-focus utterance representation technique and provided an end-to-end solution specializing in controversy focus based debate summarization (CFDS) via joint learning. Wang et al. (2020)

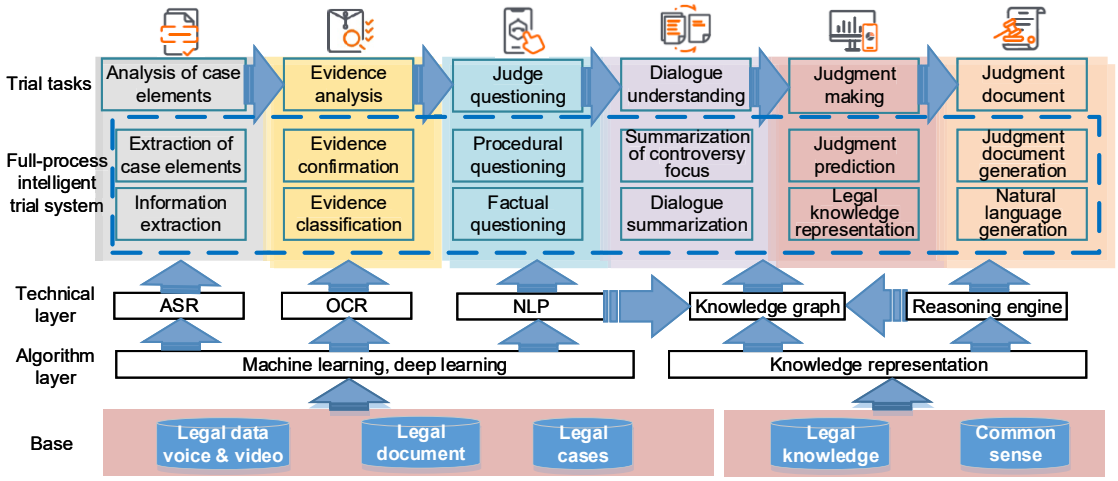

Fig. 1 Overview of the full-process intelligent trial system (FITS) (ASR: automatic speech recognition; OCR: optical character recognition; NLP: natural language processing)

investigated dialogue context representation learning with various types of unsupervised pretraining tasks, where the training objectives were given naturally according to the nature of the utterance and the structure of multi-role conversation. Wu et al. (2020) proposed a novel attentional and counterfactual natural language generation (AC-NLG) method, in which counterfactual decoders were employed to eliminate the confounding bias in data and generate judgment-discriminative court views by incorporating a synergistic judgment predictive model. Ji et al. (2020) proposed a novel network architecture, cross copy networks (CCNs), for content generation by simultaneously exploring the logical structure of the current dialogue context and similar dialogue instances.

FITS is designed by following the trial process and by emulating the way by which the judge makes judicial decisions. We adopt a combination of the knowledge-guided method and the big data driven method. The knowledge-guided method is to simulate judges based on knowledge and use logical reasoning to make judgments. The big data driven approach is to simulate judges to make judgments based on the principle of "treating like cases alike." Most of the technologies in these papers directly serve the FITS. Many new technologies were born in developing this system, and their original purpose was to perform the judicial tasks in trial practice. FITS applies these technologies to reengineer the existing case trial process and promote the intelligence of all nodes of the judicial process. In practice, FITS also provides judges and parties with intelligent assisting services at each node of the case trial procedure. Based on these works, we will show the operation process of the intelligent trial system. To summarize, we make several noteworthy contributions as follows:

1. We are the first to propose an FITS that serves primary phases of the trial procedure in the smart court.

2. We convert central judicial tasks of the trial procedure into corresponding natural language processing (NLP) problems, and adopt a combination of knowledge-based models and data-driven models.

3. Based on our FITS, we have developed an artificial intelligence (AI) judge assistant robot called Xiaozhi (micro intelligence) and achieved satisfactory results that have already assisted several courts in Zhejiang Province in financial lending cases and private lending cases.

The rest of this paper is organized as follows: In Section 2, we introduce a BiLSTM-CRF neural architecture and use it for legal text (indictments, judgment documents, etc.) information extraction. In Section 3, we justify the validity of evidence based on historical data and logical knowledge graphs. In Section 4, we propose an automatic questioning system to help judges ask procedural and factual questions. In Section 5, we summarize the focuses of the dispute during a trial by employing a multi-task learning framework called CFDS and propose a framework of dialogue inspectional summarization (DIS). In Section 6, we combine first-order logic and deep neural networks to discover the facts of the case. In Section 7, we propose the AC-NLG method to generate the court's judgment-discriminative view. In Section 8, we introduce the results achieved by FITS in the application to smart court. Section 9 discusses related research work and the last section concludes this paper.

2 Information extraction from legal documents

Information extraction (IE) aims to extract structured information from unstructured documents. It has been explored extensively due to its significant role in NLP. Legal information extraction includes the extraction of legal ontology, legal relations, and legal named entities. Earlier research studied the extraction of legal case information (Jackson et al., 2003), and combined information retrieval and machine learning to extract the correlation between current cases and precedent texts using support vector machine (SVM) and other algorithms. The transfer learning approach (Elnaggar et al., 2018) using a neural network has been trained for linking of named entities to legal documents. Recently, the popular neural structure for IE, BiLSTM-CRF (Lample et al., 2016), has shown excellent performance on numerous sequence-labeling tasks with high robustness and low computational complexity. We have collected more than 70 million judgment documents to build the corpus, including more than 360 000 court records and more than 100 000 evidence samples.

2.1 BiLSTM-CRF

The model of long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) is a type of recurrent neural network (RNN) architecture, in conjunction with an appropriate gradient-based learning algorithm, which addresses the vanishing/exploding gradient problem of learning long-term dependencies by introducing a memory cell with self-connections that store the temporal state of the network. Although numerous LSTM variants have been described, we employ the version proposed by Google (Sak et al., 2014). LSTM takes input as a sequence of vectors $x=(x_1, x_2, ..., x_n)$ and returns another sequence $y=(y_1, y_2, ..., y_n)$; then the network can be calculated using the following equations iteratively:

$$f_t = \sigma(W_{fx}x_t + W_{fm}m_{t-1} + W_{fc}c_{t-1} + b_f),$$

$$i_t = \sigma(W_{ix}x_t + W_{im}m_{t-1} + W_{ic}c_{t-1} + b_i),$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_{cx}x_t + W_{cm}m_{t-1} + b_c),$$

$$o_t = \sigma(W_{ox}x_{t-1} + W_{om}m_{t-1} + W_{oc}c_t + b_o),$$

$$m_t = o_t \odot \tanh(c_t),$$

where W is the weight matrix, b is the bias vector, σ is the logistic sigmoid function, and i, f, o, and c are, respectively, the input gate, forget gate, output gate, and cell activation vectors, all of which are of the same size as the cell output activation vector m, and $\odot$ is the element-wise product of the vectors.

The LSTM model takes past information into account, but ignores future information, because conventional RNNs are only able to make use of the previous context. Bidirectional LSTM (BiLSTM) can better exploit context in forward and backward directions. BiLSTM (Graves and Schmidhuber, 2005) combines bidirectional RNNs (BRNNs) with LSTM. BRNNs present each training sequence forward and backward to two separate recurrent nets by processing the data in both directions with two separate hidden layers that are fed forward to the same output layer. The hidden state of BiLSTM at time t generates the forward hidden sequence $\overrightarrow{h_t}$ and the backward hidden sequence $\overleftarrow{h_t}$.

A popular probabilistic method for structured prediction, conditional random fields (CRFs), is widely applied in segment and label sequence data. The advantage of CRFs is to avoid a fundamental limitation of maximum entropy Markov models (MEMMs) based on directed graphical models (Lafferty et al., 2001). We describe the definition of a general CRF (Sutton and McCallum, 2007) based on a general factor graph. Let G be a factor graph over X and Y. Then (X, Y) is a conditional random field if for any value x of X, the distribution $p(y|x)$ factorizes according to G. If $F=\{\Psi_a\}$ is the set of factors in G, then the conditional distribution for a CRF has the form

$$p(y \mid x) = \frac{1}{Z(x)} \prod_{a=1}^{A} \exp\left(\sum_k \theta_{ak} f_{ak}(y_a, x_a)\right),$$

where A is the number of factors in the collection, both feature functions f_{ak} and weights θ_{ak} are indexed by factor index a to emphasize that each factor has its own set of weights, and $Z(x)$ is a normalization factor over all state sequences for sequence x.

BiLSTM-CRF is a widely adopted neural architecture for sequence labeling problems, including entity recognition. It is a hierarchical model, and the architecture is illustrated in Fig. 2. The network can effectively obtain two-way input features through the BiLSTM layer and sentence-level tags through the CRF layer. Note that the CRF layer has a state transition matrix as a parameter, and we can effectively use past and future tags to predict the current tag.

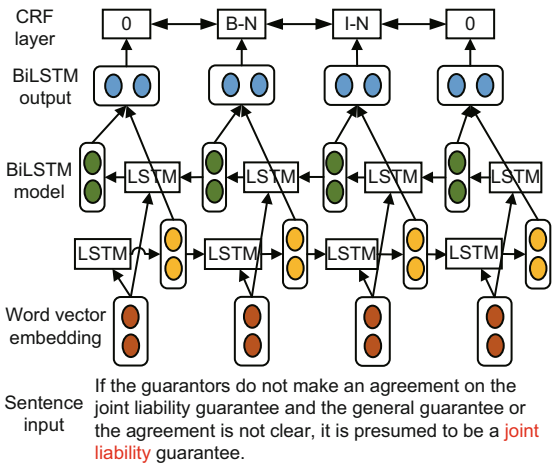

Fig. 2 The structure of BiLSTM-CRF

The first layer of the model maps words to their embeddings. $X=(x_1, x_2, ..., x_n)$ is a sentence composed of n words in a sequence, regarded as input to a BiLSTM layer. In the second layer, word embeddings are encoded and the output is

$h=(h_1,h_2,...,h_n)$. We record the features extracted from the linear layer as matrix $P=(p_1,p_2,...,p_n)$, in which the element p_{ij} corresponds to the score of the j^{th} tag of the i^{th} word in a sentence. We introduce a tagging transition matrix T, where T_{ij} represents the score of transition from tag i to tag j in successive words. The score of the sentence X along with a sequence of predictions $Y=(y_1,y_2,...,y_n)$ is then given by the sum of transition scores and network scores:

$$\text{score}(X)=\sum_{i=0}^{n-1}T_{y_i,y_{i+1}}+\sum_{i=1}^{n}P_{i,y_i}. \tag{1}$$

A softmax for all tag sequences obtains the normalized probability:

$$p(y|X)=\frac{\exp(\text{score}(X,y))}{\sum_{y'\in Y_X}\exp(\text{score}(X,y'))}, \tag{2}$$

where Y_X represents all possible tag sequences for a sentence X. The model is trained by maximizing the log-probability with a log-likelihood function (Lample et al., 2016). From this, BiLSTM-CRF obtains the sequence of output tags. In decoding the prediction, we seek the optimal path to obtain the maximum score driven by

$$y^*=\arg\max_{y'\in Y_X}\text{score}(X,y'). \tag{3}$$

Domain adaptation maps the source domain with the label and the target domain with different data distributions to the same feature space (embedding manifold). BiLSTM-CRF is combined with domain adaptation to explore external datasets (Zhao et al., 2018), as illustrated in Fig. 3, in which the full-connection layer maps the distributed feature representation to the sample label space. The CRF features can be computed separately, i.e., $\phi^{\mathcal{T}}(x)=G^{\mathcal{T}}\cdot h$, $\phi^{\mathcal{S}}(x)=G^{S}\cdot h$ for the target and source datasets, respectively. The loss functions $p\left(y\mid x;\theta^{\mathcal{T}}\right)$ and $p\left(y\mid x;\theta^{\mathcal{S}}\right)$ are optimized in alternating order.

2.2 Multi-task BiLSTM-CRF for IE

BiLSTM-CRF has been widely used in neural entity recognition (Lample et al., 2016; Liu XJ et al., 2018) and information extraction (Yang ZL et al., 2017; Zhao et al., 2018) in the legal domain. FITS applies it to the financial lending case and the private lending case. Taking the financial lending case as an example, the coverage of the extraction includes 45 types of documents (loan contract, loan extension contract, guarantee contract, mortgage contract, credit contract, pledge contract, pledge registration certificate, joint repayment commitment documents, loan vouchers, guarantor industrial and commercial registration materials, etc.), involving about 550 kinds of elements (plaintiff, defendant, defendant's ID card, litigation claims, facts and reasons, loan amount, loan contract number, signing date, the content of the indictment, etc.). On average, there are at least seven elements (fields) for each document to be extracted.

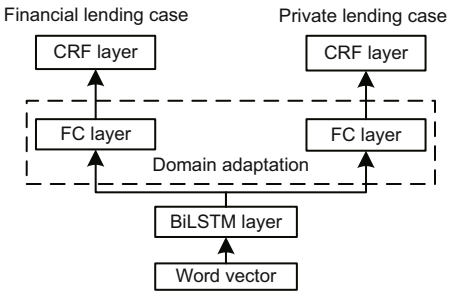

Fig. 3 Transfer learning model (Zhao et al., 2018)

The BiLSTM-CRF model first matches each input character to a word vector that is pre-trained on a large corpus (usually based on word2vec, Glove, BERT, and other language models). Then the model uses BiLSTM to perform encoding on the word vector sequence, and obtains BiLSTM word encoding after concatenation. BiLSTM word encoding is used as the top CRF layer input to obtain the final result of the beginning, inside, and outside (BIO) information identification, thereby obtaining the result of information extraction. For the example in Fig. 2, the information "joint and several liabilities" in the input will be marked and extracted. Meanwhile, many original materials are obtained through optical character recognition (OCR) or automatic speech recognition (ASR). Missing information and noise exist in the recognition process, so we use regularization rules to extract some particular information fields as a supplement.

In practice, we divide all information into two categories: general fields and specific fields. General fields refer to fields that are included in every case, such as party information. Specific fields are fields unique to each case, such as the date of contract signing for financial loan cases. For any case, general

fields will be extracted by a common model shared by all cases, and the corresponding proprietary model will extract the specific fields for this type of case. In other words, a legal case text will be extracted by two models to extract corresponding fields.

To avoid supervised learning that requires a large amount of data annotation, we also adopt the transfer learning method. We use the annotation data of one case reason to improve the information extraction ability of another case reason from transfer learning. The diagram of the migration learning model for a "financial lending case" and a "private lending case" is shown in Fig. 3. The model adds a fully connected layer (FCL) under different domains between the BiLSTM layer and the CRF sequence output layer, thereby enhancing the model's transfer learning ability.

3 Evidence analysis

In the trial, evidence analysis plays an essential role in determining the facts of the case. The primary task is to classify the evidence, which aims to divide each piece of evidence into different categories, and its purpose is to study the characteristics of different types of evidence and its application rules. The evidence materials discussed here are texts or images (for example, evidence in private lending cases includes loan agreements, guarantee conditions, payment delivery, repayment conditions, etc.). The second task of evidence analysis is to justify each piece of evidence's authenticity, legality, and relevance. These three aspects determine whether the evidence is probative.

3.1 Evidence classification

We classify different types of evidence through multi-modal analysis. The preliminary work of evidence classification is to extract text evidence from the original evidence materials through OCR technology. We then use the NLP engine to understand the text content and extract the semantic features at the text level. For the part of the evidence materials from which OCR cannot identify or accurately extract useful information, we introduce the method of visual feature recognition to improve the effect of evidence recognition. The text features and visual features are merged to classify the evidence finally. For simplicity, we here introduce mainly the classification of the evidence after it is extracted as text.

We propose a classifier by representing the evidence in a vector. Specifically, we employ the BiLSTM model introduced in the previous section to build a classifier to perform evidence classification. We apply a hierarchical attention network (Yang ZC et al., 2016) for evidence classification. The model constructs a hierarchical structure of "word-sentence-evidence text" and has two attention-level mechanisms applied at the word- and sentence-level. We learn from the idea that the model uses the attention mechanism twice under the hierarchical structure. We embed evidence in a vector representation by first using word vectors to represent sentence vectors and then using sentence vectors to represent evidence vectors.

We first encode words by embedding the words in vectors through a matrix W, and then use the BiLSTM model to obtain annotations of words by summarizing information from both directions. Afterward we obtain an annotation for a given word w by concatenating the hidden state $h=\left[\overrightarrow{h}; \overleftarrow{h}\right]$, which summarizes the information of the whole sentence centered around w. Then we introduce the attention mechanism to extract words that are important to the meaning of the sentence and aggregate the representation of those informative words to form a sentence vector. We have $u^w=\tanh(W_w h + b_w)$ as a hidden representation of h and obtain a normalized importance weight α through a softmax function. We have the sentence vector as a weighted sum of the word annotations based on the weights.

After we have the vector of the sentence, we further similarly obtain a vector of evidence. We also use BiLSTM to encode the sentences, again use attention mechanism and introduce a sentence-level context vector $u^s=\tanh(W_s h + b_s)$, and then have $v=\sum_i \alpha_i h_i$, which indicates the evidence vector that summarizes all the information of sentences in the evidence text. The evidence vector v is a high-level representation of the evidence and can be used as features for evidence classification:

$$p = \mathrm{softmax}(Wv + b). \tag{4}$$

An overview of evidence classification is shown in Fig. 4. Evidence analysis also contributes to the formation of the evidence chain, which can visually show the case fact structure. This helps the judge sort out the details of the case and grasp the trial's

progress. Evidence confirmation ensures that every piece of evidence in the evidence chain is legal and credible. Evidence classification automatically identifies different types of evidence and provides structured input for the components of the evidence chain.

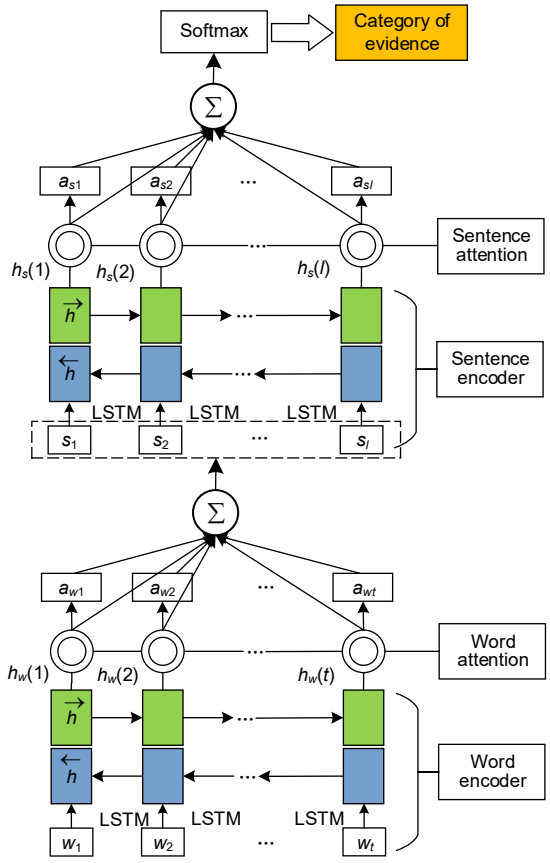

Fig. 4 Architecture of the hierarchical attention network in evidence classification

3.2 Evidence justification

The justification of evidence is the prerequisite of legal reasoning and fact-finding. The attributes of evidence are reflected in three aspects: (1) authenticity of the evidence, including authenticity of the form and authenticity of the content; (2) legality of the evidence, including legality of the source and legality of the state; (3) relevance of the evidence, that is, whether the evidence is related to the facts to be proved. Two novel methods are proposed to characterize these three attributes.

First, we evaluate the authenticity and the legality of evidence based on the analysis of historical data. In practice, it is not appropriate to determine the attributes of evidence from the legal text itself. The judge determines the authenticity and legality of evidence depending on the state of the evidence and the procedure of obtaining the evidence. The technical proposal is to mine massive evidence materials from real cases and then to calculate the prior probabilities of certain types of materials. On this basis, we build a knowledge base composed of different kinds of evidence with prior probability. According to the relevant evidence in the historical data, we evaluate the attributes by adopting the Bayesian theory to assess the probability that the evidence is real or legally obtained.

Second, we evaluate the relevance of evidence by analyzing the relationship between evidence and relevant knowledge. We adopt a logical knowledge graph based reasoning method to automatically determine the relevance of evidence. For example, in response to the "financial borrowing case," we sort out the correlations between various types of evidence based on the judge's experience and form a logical map of correlation review. For all relevant evidence materials, if there is a direct or indirect relevance between the elements of any two sets of evidence, we believe that the evidence's relevance is valid. We apply a logical graph $G=< E, R >$ to represent the relevance of evidence, where E is a set of nodes representing the type of evidence, and R is a set of links representing the relationship between two pieces of evidence.

4 Automatic questioning in trail

During the trial process, we design an automatic questioning robot to assist the judge in presiding over the trial. The trial is a particular multi-agent dialogue situation. The participants include the judge, the plaintiff, and the defendant. The judge is the trial organizer, while the plaintiff and the defendant ask questions to understand the facts. They also need to maintain order in the court trial and promote the trial process.

The automatic questioning system for the judge contains multiple modules: First, the judge's original speech is converted into text with ASR, and then the text is transformed into the context and state of the questioning system with semantic understanding. Second, a module for question management (QM) is constructed and the candidate questions are generated within this module. Finally, automatic

questioning is realized with a text-to-speech (TTS) technique that transforms the text into speech. According to the question's content, we divide the judge's questions into two categories: procedural questioning and factual questioning.

4.1 Procedural questioning

Procedural questioning refers mainly to some relatively fixed questions used by judges to organize and promote court trials, such as "identity information of the plaintiff and the defendant" and "the plaintiff and the defendant read the indictment and the defense." Procedural questioning is closely related to the procedures of the trial procedure, which has strong regularity. The system of procedural questioning focuses on solving the problem of questioning automatically in the trial procedure. The following is a sequence diagram (Fig. 5) of an automatic questioning system, where fact stands for the node of factual questioning, while procedure identifies the node of a procedural questioning node. Factual questioning is inserted in the process of procedural questioning, and multiple fact nodes can be inserted. It can be seen that an essential function of process questioning is state management.

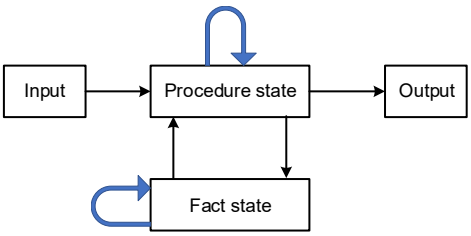

Fig. 5 Finite state machine of the questioning system

The process of procedural questioning can be defined as a natural language generation problem, and the solution includes rule-based methods and abstract generation methods. The rule-based approach has the advantages of accuracy and practicability, but it requires a large number of custom rules. The abstract generation method currently has technical bottlenecks; the generated text usually has incomplete speech, repetition, and faulty speech. The automatic questioning system innovatively proposes a scheme combining a finite state machine (FSM) and an affair map. The finite state machine is responsible for state management, and the affair map is responsible for selecting subsequent actions, which can also flexibly configure templates for downstream text generation.

4.2 Factual questioning

The judge's factual questioning is aimed mainly at the factual elements of the plaintiff's and defendant's petitions and defenses and also refers to the factual questions that the judge has asked before. Factual questioning is considered to be a text-generated problem. We obtain factual questions raised by the judge in the trial's historical dialogues, using joint learning of classification and retrieval. Therefore, we first need to define dialogue in the trial and then give an encoder to delicately represent the hierarchical information in the dialogue context.

Let $D=\{U_1,U_2,...,U_n\}$ denote a dialogue containing n utterances, where each utterance U_i is composed of a sequence of words (namely sentence) S_i, which means the text content of U_i. We employ BiLSTM to encode the semantics of the utterance. BiLSTM has been widely recognized for encoding the utterance's semantics while maintaining its syntax (Wang et al., 2020). We use BiLSTM to learn a feature representation of dialogue by masking and recovering its unit elements, such as evidence and laws in the legal domain for trial dialogue.

In the utterance layer, the input source is a set of dialogue information obtained from the speech-transformation of the judge's factual questions, denoted as a sequence of $\{\text{utterance}_1,\text{utterance}_2,...,\text{utterance}_n\}$, and each utterance is composed of the questions asked by the judge. It contains L utterances where each utterance U_i is composed of a sequence of l words (namely sentence) $S_i=\{w_{i1},w_{i2},...,w_{il}\}$ and the associated role (the judge) r_i. We employ BiLSTM to encode the semantics of the utterance. Note that the judge's role information should be embedded in the utterance. We connect the judge's role information with each word in the sentence so that the same word can be projected into different dimensional spaces. The representation of BiLSTM is obtained by concatenating its left and right context representations.

$$h_{ij}=\left[\overrightarrow{\text{LSTM}^{\text{U}}}(e_{ij});\overleftarrow{\text{LSTM}^{\text{U}}}(e_{ij})\right],j=1,2,...,l,$$

where $e_{ij}=[w_{ij};r_i]$. To strengthen the relevance between words in an utterance, the attention mechanism is employed to obtain U_i, which can be

interpreted as a local representation of an utterance:

$$U_i = \sum_{j=1}^{l} \alpha_j^u h_{ij},$$

$$\alpha_j^u = \frac{\exp(Q^u h_{ij})}{\sum_{j'=1}^{l} \exp(Q^u h_{ij'})},$$

where Q^u are learnable parameters.

In the dialogue layer, to represent the global context in the dialogue, we use BiLSTM again to encode the dependencies between utterances and obtain a global representation of each utterance, which is expressed as $\overline{U}_i$.

$$\overline{U}_i = \left[\overrightarrow{\text{LSTM}^{\text{D}}}(U_i); \overleftarrow{\text{LSTM}^{\text{D}}}(U_i)\right], i = 1, 2, ..., L,$$

$$\overline{U} = \left\{\overline{U}_1, \overline{U}_2, ..., \overline{U}_L\right\} \in \mathbb{R}^{L \times 2\text{dim}_h},$$

where dim_h refers to the dimensionality of the hidden state h.

We next perform word segmentation on the judge's utterance in the dialogue and word vector representation for each word segment to obtain $X=\{x_1, x_2, ..., x_n\}$, and then employ BiLSTM and other neural network units to encode X and conduct automatic feature selection. Because the judge's question in the dialogue contains many utterances, it therefore generates a new vector sequence $V^J=\{v_1, v_2, ..., v_n\}$. We further use the attention mechanism to perform a secondary representation of V^J. These neural network units can enhance information interaction between different levels of dialogue. After the hierarchical representation, we obtain a mapping from V^J to $V_h^J=\{h_1, h_2, ..., h_n\}$, where v and h have a one-to-one correspondence.

Because the judge's factual questions are related to the case's facts in the dialogue between the plaintiff and the defendant, it is also necessary to segment the plaintiff's litigation request and the text of the defendant's defense. We first represent the word vector for each word segment to obtain $Y=\{y_1, y_2, ..., y_n\}$. We then employ the attention mechanism to encode Y to form an encoding vector V^W for each combination. The function of V^W is to encode the information of the plaintiff's request and the defendant's defense in the encoded text. We combine the element y in V^W and the element h in V_h^J one by one according to the serial number, and the combination result is recorded as $V_h^J=\{h_1^t, h_2^t, ..., h_n^t\}$. The new statement contains the prosecution and defense information of the plaintiff and the defendant and contains information about the judge's questions in the dialogue.

We employ a classification task to recommend the most likely problem categories. We first predefine a number of problem categories. Under each question category, there are several standard question templates. For example, "recovery of debt" and "the spouses' joint debt" belong to different question categories. When recommending questions to the judge, the system obtains the indictment and pleading, as well as the historical questions raised by the judge as input, and returns the top K most likely question categories according to the steps as mentioned earlier. Finally, in the top K question categories, it returns the standard question template with the highest probability. An example of factual questioning is shown in Fig. 6.

The intelligent voice prompt system	
Procedure questioning	
Judge	Can the plaintiff and the defendant guarantee that all the above statements are true? If there are false statements, you need to bear the legal consequences of the false litigation. Have you heard clearly?
Factual questioning	
Judge	How was the plaintiff's loan delivered?
Judge	How much did the plaintiff borrow from the defendant?
Judge	Can the plaintiff guarantee the authenticity of the evidence provided by the IOU?
Judge	Except for the loan relationship in this case, do the plaintiff and defendant have any other economic relationship?

Fig. 6 An example of the questioning system

5 Trial summarization

Trial summarization consists of two tasks. The first task is to summarize the court debate transcript during the trial stage. The other task is to summarize the controversial focuses of the dialogue in the trial. Summarization-based algorithms have enabled a broad spectrum of applications, such as auto-abbreviated news and retrieval outcomes (Gerani et al., 2014) to assist users in consuming lengthy documents effectively. Thanks to the development of ASR techniques, dialogue summarization (Goo and Chen, 2018; Liu CY et al., 2019) has also attracted much attention in recent years, with exemplar applications like the judicial trial, customer service, and meeting summarization. Different from the plain document, multi-role dialogue is more

complicated due to the interactions among various parties. Enhanced representation of the atomic components (e.g., utterance and role) of the dialogue prequalifies summary generation optimization.

5.1 Summarization of controversy focus

During the trial process, the judge needs to discover the common focus of the dispute between the plaintiff and the defendant in the debate and identify how the two sides defend and refute the other party's arguments. The summary of the dialogue during the trial is vital in helping the judge grasp the critical information in the dialogue between the two parties. They include both useful information that appears during the dialogue (for example, private lending cases include the names of the parties, loan amounts, repayment records, etc.) and the focal point of the case (for example, the fact that both parties have repeatedly defended and questioned). The judge finally completes the case trial by analyzing the focus of the dialogue between the two parties and combining the judgment logic.

We have realized the automatic generation model of court trial abstracts in the intelligent trial system, mainly the automatic abstracts of dispute focuses. This task includes (1) extracting dialogue fragments related to the dispute focus in the dialogue and (2) classifying the dispute focus corresponding to each dialogue. Through the generation and processing of the court trial summary, the judge can obtain important dispute fragments in the court trial dialogue, to understand and deal with the court trial more efficiently.

The Alibaba Group proposed a multi-task learning framework called CFDS (Duan et al., 2019; Wang et al., 2020) to summarize the focus of court disputes, which includes mainly the following parts: (1) Using a sequence encoder, we model the text of the trial, semantic information of dispute focus, the role related to utterances, and the node sequence in the corresponding legal knowledge graph, and obtain the vector representation of context information through an attention mechanism. (2) According to the different dispute focuses, the focus classifier takes the category of the dispute focus involved in each utterance as the target, and obtains the label of the dispute focus. (3) For the court record summary extraction task, the objective of the summary extraction classifier is whether each utterance is extracted.

We adopt a multi-task learning strategy including the following parts: (1) the prediction of the controversy focus, (2) the highlighted sentence, and (3) the recognition of sentence elements. To distinguish between different roles in the dialogue, such as judge, plaintiff, and defendant, we use different embeddings to represent different roles. We apply word embedding to express an utterance in the dialogue through a convolutional neural network and pooling mechanism, and then use a CNN with an attention mechanism to express the entire dialogue. The process of trial summarization is shown in Fig. 7.

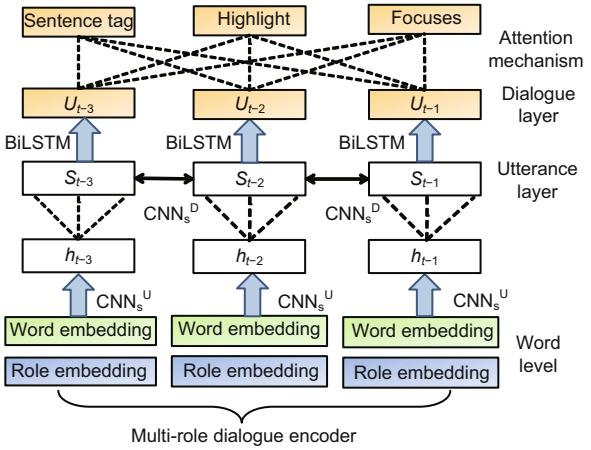

Fig. 7 The process of trial summarization

5.1.1 Controversy focus assignment

The first task is to assign a controversy focus to each utterance. Different debate dialogues may have various controversy focuses, and the judge concludes each controversy focus according to the content of debate dialogue D. Because the number of controversy focuses varies in different debate dialogues and each controversy focus differs in semantics and syntax, we can hardly cope with this task using text classification. We calculate the relevance between utterance u_i and each controversy focus f_m in F with respect to debate dialogue D.

To do so, we need to compute the embedding of each controversy focus. As both controversy focuses and sentences in the debate are natural language, we use the BiLSTM encoder to obtain the controversy focus embedding f_m. In addition, not every utterance u_i is assigned a controversy focus. Some utterances do not belong to any controversy focus and they can be regarded as irrelevant content, namely

noise. Thus, a category Noise is created for every debate dialogue and a dense vector is used to represent it. Then we calculate the attention score α_{ij}^{f} of utterance u_i with f_j:

$$\alpha_{im}^{f}=\frac{\exp(u_i^{\mathrm{T}}\cdot W^{f}\cdot f_m)}{\sum_{m=1}^{M+1}\exp(u_i^{\mathrm{T}}\cdot W^{f}\cdot f_m)}. \tag{5}$$

Controversy focus with the highest normalized score α_{ij}^{f} is the controversy focus assigned to u_i.

5.1.2 Utterance extraction

The second task aims to extract the crucial utterances from the debate dialogue about the different controversy focuses and to form multiple summarizations. The utterance extractor considers two aspects: utterance content and controversy focuses. To enhance utterance representation learning, we employ the normalized controversy focus distribution as the input to this task:

$$F_i=\sum_{m=1}^{M+1}\alpha_{im}^{f}\cdot f_m. \tag{6}$$

Then F_i and u_i are concatenated and fed into the fully connected layers as follows:

$$o_i=\mathrm{sigmoid}(W_2^{f_c}\cdot\mathrm{ReLU}(W_1^{f_c}\cdot[F_i,u_i])), \tag{7}$$

where $W_1^{f_c}$ and $W_2^{f_c}$ are two weight matrices and $o_i\in[0,1]$ is the output of the utterance extractor, which indicates the probability of extracting utterance u_i.

5.2 Dialogue inspectional summarization

In the court debate scenario, the judge summarizes the case narrative based on facts recognized from the court debate during the trial and relies on the evidence or materials submitted by the litigants. We particularly propose a framework of DIS, which includes four parts: (1) For the text of the trial transcript, the multi-role dialogue encoder can hierarchically and serially model the semantics of the court trial transcript, and obtain the vectorized representations of the word level, speech level, and dialogue level, respectively. (2) The decoder uses the attention mechanism and the replication mechanism to generate the sequence results identified by the court. (3) The target fact element regularizer classifies the relevance of fact elements, and the element level in the generated text should be consistent with the content of the court trial. (4) The missing fact entity discriminator uses the classification of missing fact entities to predict the inconsistency between the decoder state representation and the dialogue encoding representation in fact entity classification.

We design a hierarchical dialogue encoder involving role information to accommodate extended context and multiple turns among the multiple roles. Rather than directly aligning the input dialogue and its summary, within the generation framework, we propose two additional tasks in the manner of joint learning: expectant factual aspect regularization (EFAR) can estimate the factual aspects to be contained in the summary to make the model emphasize the factual coverage of logical reasoning, and missing factual entity discrimination (MFED) predicts the missing aspects, which discover/alarm the factual gap between the input and the output. Specifically, the DIS framework is shown in Fig. 8.

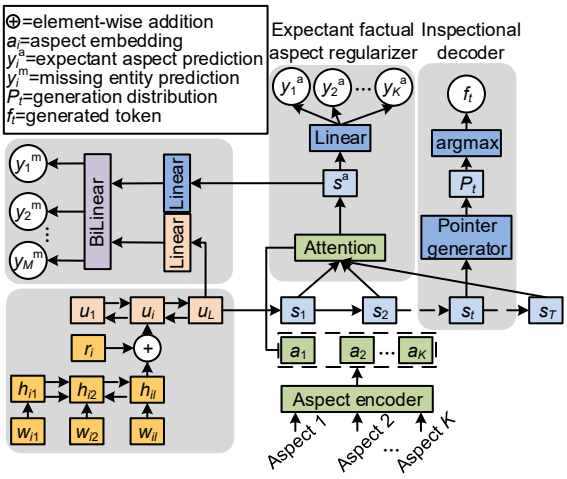

Fig. 8 Overview of the dialogue inspectional summarization (DIS) framework

5.2.1 Inspectional decoder

We propose an inspectional decoder for generating summaries. The inspectional decoder generates the summary via a pointing mechanism, while the expectant factual aspect regularizer ensures factual consistency from the aspect level.

From the perspective of bionics, humans tend to write a draft before focusing on factual aspects. We treat the inspectional decoder as a drafter, whose states need to be further regularized by the aspect-aware module.

With the pointing mechanism integrated, the decoder can directly copy tokens from dialogue,

making the generated summary more accurate and relevant in factual details.

5.2.2 Expectant factual aspect regularizer

When writing formal documents like the legal verdict, people always carefully review their drafts to ensure that there are no inconsistencies in the expected aspects. Inspired by this process, we propose an expectant factual aspect regularizer to verify the aspect level's consistency.

For each aspect e_i, we use the aspect encoder to obtain its semantic embedding a_i. The encoder $\mathrm{Enc}^{\mathrm{A}}$ is single-layer bidirectional LSTM to represent the aspect description text:

$$a_i = \mathrm{Enc}^{\mathrm{A}}(e_i). \tag{8}$$

We then produce a weighted sum of the decoder hidden states, known as the aspect-aware decoder state s^{a}:

$$\begin{cases} s^{\mathrm{a}} = \frac{1}{K}\sum_{i=1}^{K}\sum_{t=1}^{\mathrm{T}} \alpha_{it}^{\mathrm{asp}} s_t, \\ \alpha_{it}^{\mathrm{asp}} = \underset{t}{\mathrm{softmax}}(\mathrm{score}(a_i, s_t)), \end{cases} \tag{9}$$

where K is the number of factual aspects and the score function uses additive attention:

$$\mathrm{score}(a_i, s_t) = v^{\mathrm{T}}\tanh(\mathrm{linear}(a_i, s_t)). \tag{10}$$

Finally, we feed s^{a} into a three-layer classifier to predict the expectant aspects:

$$y^{\mathrm{a}} = \sigma(\mathcal{F}^{\mathrm{a}}(s^{\mathrm{a}})), \tag{11}$$

where $\mathcal{F}^{\mathrm{a}}$ is the notation of linear layers and $y^{\mathrm{a}} \in \mathbb{R}^K$ indicates the related probability of K aspects.

5.2.3 Missing factual entity discriminator

There are always factual inconsistencies between the dialogue and reference summary. In the Seq2Seq framework, inconsistencies mislead the decoder to generate incorrect factual details. The missing factual entity discriminator tries to detect the inconsistencies, thus mitigating the problem. Motivated by this observation, we design the discriminator to classify whether the factual entity is missing in the conversation. In real applications, human summarizers can refer to the predictions to complete generated text based on additional information. Intuitively, we view inconsistency as the factual divergence between source and target content, using the bilinear layer as the classifier.

6 Judgment prediction

Legal judgment prediction (LJP) is one of the most attractive research topics in the field of legal AI (Xiao et al., 2018; Chao et al., 2019; Zhong et al., 2020a, 2020b). LJP aims to predict legal judgment based on a legal text including the description of the case facts. Most previous works treated LJP as a text classification task and generally adopted DNN-based methods to solve it. Zhong et al. (2018) and Yang WM et al. (2019) used multi-task learning to capture the dependencies among subtasks by considering their topological order. Zhong et al. (2020b) applied a question-answering task to improve the interpretability of LJP through reinforcement learning. Luo et al. (2017) formulated legal documents as a knowledge basis and used attention mechanisms to aggregate representations of relevant legal texts to support judgment prediction.

We combine DNNs with a symbolic legal knowledge module, in which legal knowledge is expressed as a set of first-order logic (FOL) rules. The application of FOL to represent domain knowledge has already demonstrated its effectiveness on many other tasks, including visual relation prediction (Xie et al., 2019), natural language inference (Li et al., 2019), and semantic role labeling (Li et al., 2020). The advantages of representing legal knowledge as FOL rules can make judgment prediction more interpretable and provide models with inductive bias, which reduces neural network dependency.

The proposed model unifies the gradient-based deep learning module with the non-differentiable symbolic knowledge module via probabilistic logic. Specifically, we build a deep learning module based on a co-attention mechanism, which benefits the information interaction between fact descriptions and claims. Afterward, the deep learning module outputs, predicted probability distribution for judgments, will be fed into the symbolic module.

6.1 Legal knowledge represented as logic rules

Before presenting how to integrate legal knowledge into DNNs, we briefly introduce FOL to express legal knowledge. To preserve the advantages of gradient-based end-to-end training schema, we convert the Boolean operations of FOL into probabilistic logic, denoted in the continuous real-valued space.

Specifically, we associate the variable X in

preconditions with corresponding neural outputs x. Then, Lukasiewicz T-norm and T-conorm (Klement et al., 2000) are used to relax the logic rules to a softened version based on the associated outputs of the deep learning module. A set of functions is denoted to map the discrete outputs of FOL into continuous real values as follows:

1. $\Gamma(X_i) = x_i$ with X_i denoting a variable in FOL and x_i as the associated output of neural networks.
2. $\Gamma(\bigwedge_i X_i) = \max(0, \sum_i x_i - |X| + 1)$.
3. $\Gamma(\bigvee_i X_i) = \min(1, \sum_i x_i)$.
4. $\Gamma(\neg \bigvee_i X_i) = \max(0, 1 - \sum_i x_i)$.
5. $\Gamma(\neg \bigwedge_i X_i) = \min(0, N - \sum_i x_i)$.

In designing qualified mapping functions, when the precondition holds, the mapping function should generate a predefined maximum positive score to lift the original score produced by neural networks. The mapping functions should also reveal the semantics of propositional connectives. For example, the conjunctive precondition's mapping score becomes zero if even only one of the conjuncts is false. For a disjunctive precondition, the mapping score becomes zero when all the disjuncts are false. Moreover, the mapping score will increase as the number of disjuncts increases.

In addition to the functions listed above, two mapping functions are used for negated predicates. One of them is for negated predicates in preconditions, e.g., $\neg X_i$. The soften output of $\neg X_i$ is denoted as $1 - x_i$. The other is for negated consequent $\neg Y$, designated as $-y_i$ to reduce neural networks' original outputs.

6.2 Three typical types of legal knowledge

We investigate compiling three specific types of legal knowledge into FOL rules, which are frequently referred to by legal experts in private loan cases.

The first legal logic rule comes from article 28 of the Supreme People's Court's Provisions on Several Issues Concerning the Application of Law in the Trial of Private Loan Cases (http://www.court.gov.cn/fabuxiangqing-15146.html). In short, it is stated that the law shall not support the interest rate agreed by the lender and the borrower exceeding four times the quoted interest rate on the one-year loan market when the contract was established. We formulate this legal knowledge as the following FOL rule K_1:

$$X_{\mathrm{TIR}} \wedge X_{\mathrm{RIO}} \rightarrow \neg Y, \tag{12}$$

where X_{TIR} is a variable that indicates if the current claim is for interest. X_{RIO} indicates if the claimed interest rate exceeds four times the quoted interest rate on the one-year loan market. This rule reflects the decrease in the illegitimate interest rate.

The second legal logic rule comes from article 29 of the same law. In short, it is stated that if neither the interest rate during the loan period nor the overdue interest rate has been agreed upon, the people's court shall support the unpaid interest from the date of overdue repayment. We formulate this legal knowledge as the following FOL rule K_2:

$$X_{\mathrm{TIR}} \wedge \neg X_{\mathrm{RIA}} \wedge \neg X_{\mathrm{DIL}} \rightarrow \neg Y, \tag{13}$$

where X_{RIA} indicates if the borrower and the lender have made an agreement on the interest rate, and X_{DIL} indicates if the date of overdue repayment is legitimate.

In private loan law cases, the plaintiff often proposes multiple claims and the judgments on these claims are not independent. For example, when a plaintiff proposes two claims, one is for the principal and the other is for the interest. If the judge does not support the principal claim, then the interest claim should not be supported either. Such prior knowledge should be injected into the deep learning module as well. Another example showing the dependency among multiple claims is that the losing party shall bear the litigation costs. The third FOL rule, K_3, is formulated as

$$\bigwedge_{j \in s, j \neq i} Y_j \wedge X_{\mathrm{TIC}} \rightarrow Y_i, \tag{14}$$

where X_{TIC} indicates if the current claim is for litigation fees or not. This rule will affect those claims for litigation costs.

6.3 Injecting legal knowledge into DNNs

We first build a co-attention network as our base model, which can enrich the representations by exchanging information between fact descriptions and claims. Formally, we provide an abstract denotation

of the co-attention network as follows:

$$\begin{cases} H_c = \text{encoder}(C), \\ H_f = \text{encoder}(F), \\ \overleftarrow{p}, \overleftarrow{q} = \sigma(\text{layers}(H_c, H_f)), \\ H_{fc} = \sigma(\text{layers}(\overleftarrow{p}, \overleftarrow{q}, H_c, H_f)), \\ y = \text{softmax}(WH_{fc}). \end{cases} \tag{15}$$

Here, the encoder and layers are deep neural networks. σ and W are the activation function and model parameters, respectively. Note that the softmax outputs of co-attention networks will be input into the logic module and adjusted accordingly.

As shown in Fig. 9, the proposed model consists of a deep learning module based on co-attention networks and a symbolic legal knowledge module. We first input fact descriptions and multiple claims in the co-attention network to obtain contextual representations for both fact descriptions and claims. The predicted probability distribution of the deep learning module is then re-weighted by first-order logic rules in the symbolic module. The logic rules represent professional legal knowledge, which is essential for making correct judgments.

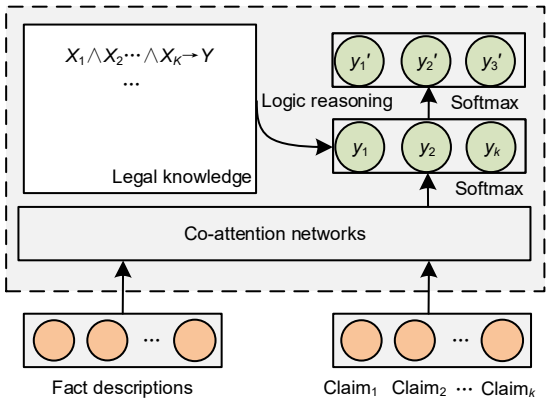

Fig. 9 The overall architecture of the proposed model

The co-attention model can fuse the claim representations and fact descriptions to create implicit reasoning. However, the related legal knowledge used by legal experts (e.g., lawyers or judges) can hardly be learned by the co-attention network. For example, the rule that a private loan interest rate that exceeds 2% per month is not protected by law may not always be followed by the neural networks. Thus, it is crucial to explicitly inject such declarative legal knowledge into neural networks, so they can make interpretable judgment predictions.

Before introducing substantial legal knowledge related to our private loan scenario, we first show how to inject symbolic FOL rules into the deep learning module using the above mapping functions $\Gamma(\cdot)$. In short, the core idea of this legal knowledge injection is to re-weight the output y of co-attention networks as introduced in the previous subsection so that when the facts in the text satisfy conditions in the legal knowledge, the associated value of y increases. Otherwise, the value of y decreases.

Specifically, given the softmax outputs y of Eq. (15) and an FOL rule $X \rightarrow Y$, the FOL rule and DNNs are combined by regulating the outputs of the deep learning module as follows:

$$y' = \text{softmax}(y + \rho\Gamma(X)), \tag{16}$$

where ρ is a hyper-parameter which denotes the importance of each rule.

Through Eq. (16), we can directly regulate the deep learning module's outputs.

Given a set of samples, $D = \{F_i|_{i=1}^N, C_i^j|_{j=1}^K\}$, the model is trained by maximizing the following objective function:

$$J = \sum_{i=1}^{N}\sum_{j=1}^{K} \ln(y'_{ij}). \tag{17}$$

7 Judgment document generation

Judgment document generation is based mainly on the judge's view, which is often regarded as a "court view" in the judgment document (Ye et al., 2018), and its content includes mostly the determination of the case facts and the matching of laws and regulations. Therefore, the core task of judgment document generation is the generation of the court's view. Details about the proposed algorithms and experimental results on court's view generation can be found from our previous conference paper published in EMNLP 2020 (Wu et al., 2020).

Due to the popularity of machine learning, especially NLP techniques, many legal assistant systems have been proposed to improve the effectiveness and efficiency of the legal system from different aspects. The court's view can be regarded as the interpretation of the sentence in a case. As an important portion of the verdict, the court's opinion is difficult to generate due to the logical reasoning required in the content. Therefore, the generation of the court's

view is regarded as one of the most critical functions in a legal assistant system. The court's view consists of two main parts, the judgment and the rationales, where the judgment responds to the plaintiff's claims in civil cases or charges in criminal cases, and the rationales are summarized from the fact description to derive and explain the judgment.

In this work, we focus on the problem of automatically generating the court's view in civil cases by injecting the plaintiff's claim and fact description (Fig. 10). In such a context, generating the court's view can be formulated as a text-to-text NLG problem, where the input is the plaintiff's claim and the fact description. The output is the corresponding court view, which contains the judgment and the rationales. Because the claims are various, for simplification, the judgment of a civil case is defined as supported if all its requests are accepted and non-supported otherwise.

Plaintiff's claim	The plaintiff A claimed that the defendant B should return the loan of $29 500 Principle claim and the corresponding interest Interest claim.
Fact description	After the hearing, the court held the facts as follows: defendant B borrowed $29 500 from plaintiff A, and agreed to return after one month. After the loan expired, the defendant failed to return Fact.
Court's view	The court concluded that the loan relationship between plaintiff A and defendant B is valid. The defendant failed to return the money on time Rationale. Therefore, the plaintiff's claim on principle was supported Acceptance according to law. The court did not support the plaintiff's claim on interest Rejection because the evidence was insufficient Rationale.

Fig. 10 An example of the court's view from a legal document (Wu et al., 2020)

Although classical NLG models have been applied to many text-generation tasks, when generating the court's view, such techniques cannot be applied for the following reasons: (1) The "no claim, no trial" principle exists in civil legal systems; the judgment is the response to the claims declared by the plaintiff, and its rationales summarize the corresponding facts. (2) The distribution of judgment results in civil cases is very imbalanced. Such an imbalance of judgment would blind the model's training by focusing on the supported cases while ignoring the non-supported cases, leading to incorrect judgment generation of the court's view.

To address these challenges, we propose the AC-NLG method by jointly optimizing a claim-aware encoder, a pair of counterfactual decoders to generate judgment-discriminative court views (both supportive and non-supportive), and a synergistic judgment predictive model. Comprehensive experiments show the effectiveness of our method under both quantitative and qualitative evaluation metrics.

7.1 Backdoor adjustment

Causal inference (Pearl, 2009; Kuang et al., 2020) is a powerful statistical modeling tool for explanatory analysis that removes the confounding bias in data. That bias might create a spurious correlation or confounding effect among variables. Recently, many methods have been proposed to remove the confounding bias in the literature of causal inference, including do-operation based on a structure causal model (Pearl, 2009) and counterfactual outcome prediction based on a potential outcome framework (Imbens and Rubin, 2015). With do-operation, a backdoor adjustment (Pearl et al., 2016) has been proposed for data debiasing. In this study, we sketch the causal structure model of our problem, as shown in Fig. 11, and adopt the backdoor for confounding bias reduction.

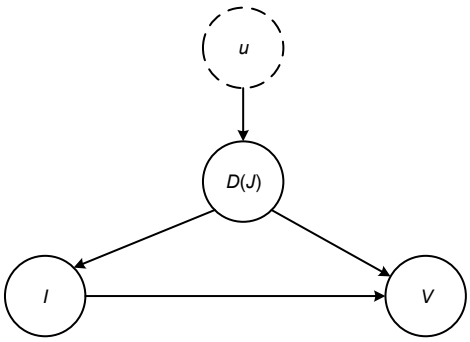

Fig. 11 Confounding bias from the data generation mechanism (Wu et al., 2020)

In this subsection we introduce the effect of mechanism confounding bias on the generation of the court's view and propose a backdoor-inspired method to eliminate that bias. Then, we describe our AC-NLG model in detail. Fig. 12 shows the overall framework.

As shown in Fig. 11, u refers to the unobserved mechanism (i.e., plaintiffs sue when they have a high probability of being supported) that causes the judgment in dataset $D(J)$ to be imbalanced. $D(J) \to I$ denotes that the imbalanced data $D(J)$ has a causal

effect on the representation of input I (i.e., plaintiff's claim and fact description), and $D(J) \to V$ denotes that $D(J)$ has a causal effect on the representation of court's view V. Such imbalance in $D(J)$ leads to the confounding bias that the representations of I and V tend to be supportive and blind the conventional training on $P(V|I)$. The confounding bias from the data generation mechanism would blind the conventional training on $P(V|I)$, and current sequence-to-sequence models struggle to solve this problem. For a particular case, given the input $I = (c, f)$, and using the Bayes rule, we would train the model to generate the court's view V as follows:

$$P(V|I) = \sum_j P(V|I, j)P(j|I). \tag{18}$$

The backdoor adjustment creates a do-operation on I, which promotes the posterior probability from passive observation to active intervention. The backdoor adjustment addresses the confounding bias by computing the interventional posterior $P(V|\mathrm{do}(I))$ and controlling the confounder as

$$P(V|\mathrm{do}(I)) = \sum_j P(V|I, j)P(j). \tag{19}$$

Because the backdoor adjustment helps cut the dependence between $D(J)$ and I, we can eliminate the confounding bias from the data generation mechanism and learn an interventional model for debiased court's view generation.

As shown in Fig. 12, to optimize Eq. (19), we use a pair of counterfactual decoders to learn the likelihood $P(V|I, j)$ for each j. At inference, we propose to use a predictor to approximate $P(j)$. Note that our implementation on backdoor-adjustment can be easily applied for multi-valued confounding with multiple counterfactual decoders.

7.2 Model architecture

Our model is conducted in a multi-task learning manner that consists of a shared encoder, a predictor, and a pair of counterfactual decoders. Note that the predictor and the decoders take the output of the encoder as input.

1. Claim-aware encoder

Intuitively, the plaintiff's claim c and the fact description f are sequences of words. The encoder first transforms the words into embeddings. Then the embedding sequences are fed to BiLSTM, producing two sequences of hidden states h^c and h^f corresponding to the plaintiff's claim and the fact description, respectively.

After that, we use a claim-aware attention mechanism to fuse h^c and h^f. For each hidden state h_i^f in h^f, e_k^i is its attention weight on h_k^c, and the attention distribution q^i is calculated as follows:

$$e_k^i = v^{\mathrm{T}} \tanh(W_c h_k^c + W_f h_i^f + b_{\mathrm{attn}}), \tag{20}$$

$$q^i = \mathrm{softmax}(e^i), \tag{21}$$

where v, W_c, W_f, b_{attn} are learnable parameters. The attention distribution can be regarded as the importance of each word in the plaintiff's claim. Next, the new representation of the fact description is produced as follows:

$$h_i^{f*} = h_i^f + \sum_k q_k^i h_k^c. \tag{22}$$

After feeding to another BiLSTM layer, we obtain the claim-aware representation of fact h.

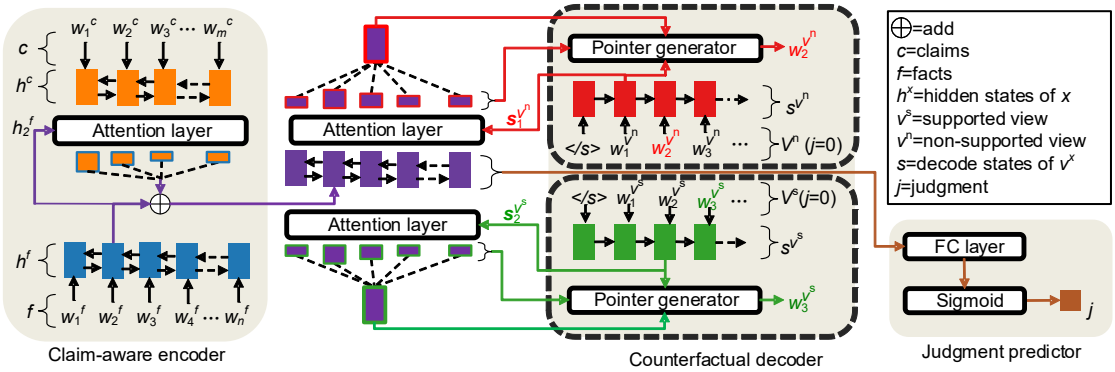

Fig. 12 Architecture of the attentional and counterfactual natural language generation (AC-NLG) method (Wu et al., 2020)

2. Judgment predictor

Given the claim-aware representation of fact h, the judgment predictor produces the probability of support P_{sup} through a fully connected layer and a sigmoid operation. The prediction result j is obtained as follows:

$$j = \begin{cases} 1, & P_{\text{sup}} > 0.5, \\ 0, & P_{\text{sup}} \leq 0.5, \end{cases} \tag{23}$$

where 1 means support and 0 means non-support.

3. Counterfactual decoder

To eliminate the effect of data bias, here we use a pair of counterfactual decoders, which contains two decoders, one for supported cases and the other for non-supported cases. The two decoders have the same structure but aim to generate the court's view with different judgments. We name them counterfactual decoders because every time only one of the two generated court views is correct. Still, we apply the attention mechanism. At each step t, given the encoder's output h and the decode state s_t, the attention distribution a^t is calculated in the same way as q^i in Eq. (21), but with different parameters. The context vector h_t^* is then a weighted sum of h:

$$h_t^* = \sum_i a_i^t h_i. \tag{24}$$

The context vector h_t^*, which can be regarded as a representation of the input for this step, is concatenated with the decode state s_t and fed to linear layers to produce the vocabulary distribution p_{vocab}:

$$p_{\text{vocab}} = \text{softmax}(V'(V[s_t, h_t^*] + b) + b'), \tag{25}$$

where V, V', b, b' are all learnable parameters. Then we add a generation probability to solve the out of vocabulary (OOV) problem. Given the context h_t^*, the decode state s_t, and the decoder's input (the word embedding of the previous word) x_t, the generation probability P_{gen} can be calculated:

$$P_{\text{gen}} = \sigma(w_{h^*}^{\text{T}} h_t^* + w_s^{\text{T}} s_t + w_x^{\text{T}} x_t + b_{\text{ptr}}), \tag{26}$$

where w_{h^*}, w_s, w_x, and b_{ptr} are learnable, and σ is the sigmoid function. The final probability for a word w in time step is obtained:

$$P(w) = P_{\text{gen}} \cdot p_{\text{vocab}}(w) + (1 - P_{\text{gen}}) \sum_{i:w_i=w} a_i^t. \tag{27}$$

We introduce how to alienate the two decoders in the training part.

4. Training

For the predictor, we use cross-entropy as the loss function:

$$\mathcal{L}_{\text{pred}} = -\hat{j}\ln(P_{\text{sup}}) - (1 - \hat{j})\ln(1 - P_{\text{sup}}), \tag{28}$$

where $\hat{j}$ is the real judgment.

For the decoders, the previous word in training is the word in the real court's view, and the loss for time step t is the negative log-likelihood of the target word w_t^*:

$$\mathcal{L}_t = -\ln P(w_t^*), \tag{29}$$

and the overall generation loss is

$$\mathcal{L}_{\text{gen}} = \frac{1}{T} \sum_{t=0}^{T} \mathcal{L}_t, \tag{30}$$

where T is the length of the real court's view.

Because we aim to make the two decoders generate two different court views, we use a mask operation when calculating the loss of each decoder. The exact loss for the support decoder is

$$\mathcal{L}_{\text{sup}} = \begin{cases} \mathcal{L}_{\text{gen}}, & \hat{j} = 1, \\ 0, & \hat{j} = 0. \end{cases} \tag{31}$$

The loss for the non-support decoder $\mathcal{L}_{\text{nsup}}$ is obtained in the opposite way. Thus, the total loss is

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{sup}} + \mathcal{L}_{\text{nsup}} + \lambda \mathcal{L}_{\text{pred}}, \tag{32}$$

where we set λ to 0.1 in our model.

8 Application and results

To investigate the effectiveness of FITS, we conducted experiments on a real private loan dataset. We developed an AI-judge assistant, named Xiaozhi, based on FITS. We also applied FITS in real courts and achieved satisfactory results.

8.1 Experiment

Due to the page limitation, here we show only the comparison results of judgment prediction, which is the most important task of a smart trial. We compare our method with other deep learning baselines on the collected private loan dataset and discuss the role that legal knowledge plays in its performance.

8.1.1 Experimental setting

We collected a total of 61 611 private loan law cases. Each instance in the dataset consists of a factual description and the plaintiff's multiple claims. We will release all the experiment data to motivate other scholars to investigate this problem further. Macro F1 and Micro F1 (Mac.F1 and Mic.F1 for short) were adopted as the primary metrics for algorithm evaluation. We denoted the co-attention-based method as CoATT+LK, which means we injected legal knowledge into neural networks.

8.1.2 Overall performance

We evaluated our model and the baselines on the private loan dataset. In addition to Mac.F1 and Mic.F1, we used macro-precision (Mac.P) and macro-recall (Mac.R) to evaluate the methods. The performance on the test set is summarized in Table 1. We can draw the following conclusions from the results: First, the performance of the deep learning based methods, e.g., TextCNN, BiLSTM+ATT, and HARNN, significantly exceeded the traditional machine learning method TF-IDF+SVM, which shows the success of applying neural networks for LJP. Second, LSTM-based methods gave better results than the CNN-based approach, demonstrating the advantages of extracting contextual features using LSTM. Third, BERT outperformed all the deep learning based methods, which shows the pre-trained language model's strong representation abilities, even for the legal domain.

Finally, the co-attention model gave a 4.8% absolute increase in performance (the average of Mac.F1 and Mic.F1) compared with BERT, which leads to two conclusions. First, directly applying pre-trained models to specific domains still has room for improvement. Second, it verifies our assumption that the bi-directional attention flows of information between facts and claims help locate crucial facts. Most importantly, injecting legal knowledge into co-attention networks gave another 1% absolute increase compared with the co-attention model and achieved the best results among all methods.

8.2 Application

The full-process smart trial system has played an important role in the construction of the smart court in Zhejiang Province. We developed a substantive AI-judge assistant robot, called Xiaozhi based on FITS, which has already assisted seven Zhejiang Provincal courts in financial lending cases and private lending cases. Xiaozhi moved the full procedural trial mode from the experimental stage to application practice. As a judge's assistant, Xiaozhi demonstrates the advantages of AI in the judicial field. FITS can understand legal documents, extract case information, justify evidence, and record the parties' speeches. It assists the judge in automatically questioning, promoting the trial process independently, summarizing the focus of disputes, predicting the outcome of the judgment, and generating judgment documents. If the judge's judgment deviates from a similar case, the system will also remind the judge of risks.

Compared with the traditional court, FITS has allowed realization of a new "human–machine integration" mode of intelligent trial in real applications. The litigation procedures in China consist of four phases: (1) In the trial preparation phase, Xiaozhi can push the pre-trial report to the judge and analyze the report's elements. (2) In the investigation stage, Xiaozhi synchronously conducts semantic recognition and text conversion, automatically helps

Table 1 Final results of all methods on the civil loan test dataset (Wu et al., 2020)

Method	Reject			Partially support			Support			Average			
	P	*R*	F1	*P*	*R*	F1	*P*	*R*	F1	Mac.P	Mac.R	Mac.F1	Mic.F1
TF-IDF+SVM	0.751	0.494	0.596	0.581	0.454	0.510	0.848	0.922	0.884	**0.805**	0.663	0.624	0.727
TextCNN	0.756	0.434	0.551	0.665	0.417	0.513	0.830	**0.945**	0.884	0.750	0.599	0.649	0.807
BiLSTM+ATT	0.722	0.528	0.609	0.645	0.512	0.571	0.858	0.926	0.890	0.741	0.655	0.690	0.818
HARNN	**0.758**	0.521	0.617	0.633	0.505	0.562	0.855	0.923	0.889	0.749	0.650	0.689	0.816
BERT	0.723	0.608	0.667	0.645	0.579	0.610	0.876	0.913	0.894	0.748	0.700	0.722	0.827
CoATT	0.705	**0.728**	0.716	**0.727**	0.690	0.708	0.914	0.923	0.918	0.782	0.780	0.781	0.864
CoATT+LK	0.750	0.695	**0.721**	0.718	**0.753**	**0.735**	**0.926**	0.921	**0.923**	0.798	**0.789**	**0.793**	**0.872**

The best results are in bold

the judge with questioning, and justifies the validity of evidence. (3) In the debate stage, Xiaozhi can convert the dialogue between the parties into text in real time, and summarize the dispute's focus from the dialogue and extract its elements. (4) In the judgment stage, Xiaozhi helps predict the outcome of the case and generate judgment documents in real time, which enables the judge to pronounce judgment in court after review and confirmation.

FITS breaks through the geographical limitations and avoids the inefficiency of traditional courts. It has launched "networking," "digitization," and "intelligence" in the smart court. The application of FITS has achieved satisfactory results: (1) In the automatic questioning task, the accuracy rate of procedural questioning can reach 96%. The hit rate for factual questioning can reach 70%. (2) In the high-frequency private lending and financial borrowing cases, the summaries of court trial records can reach 90%, and the accuracy rate of generating dispute focuses can reach 70%. The factor prediction accuracy rate can reach 80%. (3) The accuracy of financial loan evidence determination is 92%, and the accuracy of private lending is 95%. The accuracy of evidence classification can reach 90%. (4) FITS predicts the trial's outcome by combining the legal knowledge graph and big data analysis, with an accuracy rate of 96%. (5) With the help of our system, the rate of sentence pronouncement in court can be improved from 40% (traditional judge system) to 90%, and the proposed system can also shorten the trial time from 2–3 h (traditional judge system) to 20–30 min. Moreover, the average number of trial days for initial financial loan cases has been shortened from 98 in 2017 to 66 in 2020, and no case has been revised or remanded for retrial.

9 Related works

This paper attempts to cover the primary process of adjudication; the essential steps/stages for a trial pipeline include making judgments and writing judgment documents. The technologies of text classification and legal prediction are often used to assist in these tasks. In the history of AI and law, there have been many research works. Basically, the legal text classifier is the fundamental technology of our work. Dahbur and Muscarello (2003) gave a classification system for serial criminal patterns. Ashley and Brüninghaus (2009) proposed a model of SMILE+IBP to automatically classify textual facts in terms of a set of classification concepts that capture stereotypical fact patterns. Passage-based text summarization was used to investigate how to categorize text excerpts from Italian normative texts (Kanapala et al., 2019). Liu CL and Chen (2019) applied machine learning methods, including gradient boosting, multilayer perceptrons, and deep learning methods with LSTM units, to extract the gist of Chinese judgments of the supreme court.

Concerning the works of legal prediction, remarkable results have been achieved (Arditi et al., 1998). In the early stages, machine learning, such as argument based machine learning (Možina et al., 2005), was applied to the legal domain. Machine learning has also been applied to predict decisions of the European Court of Human Rights (Aletras et al., 2016; Medvedeva et al., 2020). A time-evolving random forest classifier was designed to predict the behavior of the Supreme Court of the United States (Katz et al., 2017). Recently, Chao et al. (2019) improved the interpretability of charge prediction systems and improved automatic legal document generation from the fact description. They further proposed an interpretable model for charge prediction for criminal cases using a dynamic rationale attention mechanism (Ye et al., 2018). Hu et al. (2020) studied the problem of identifying the principals and accessories from the fact description with multiple defendants in a criminal case.

10 Conclusions

This paper presents a full-process intelligent trial system. The technical route adopts mainly a combination of knowledge-based models and data-centric models. The method of knowledge expression and reasoning formalizes mainly the judge's legal knowledge and implements logical reasoning according to the judge's logical rules. Big data driven technology realizes the tasks of classification, summarization, and prediction through big data analysis of massive legal texts. Several deep learning models are proposed for legal information extraction, evidence justification, trial summarization, outcome prediction, and judgment document generation.

Note that the application of FITS has not been extended to criminal cases. The application to

criminal cases should be very cautious because the standard of judicial proof in criminal cases is "beyond a reasonable doubt," but the prediction results of the intelligent system cannot be guaranteed to be 100% correct. The predictive model contains machine learning algorithms that are uninterpretable or have "black box" problems, which means that the process from data input to result from output is nontransparent. Therefore, the use of FITS in criminal case trials will be very cautious.

The system explores the in-depth application of big data, modern logic, and AI in the full trial process. The AI trial system also has shortcomings. Even if the existing technologies are good at handling simple cases (such as financial lending and private lending cases), for complex cases, the determination of the facts of the case and the application of laws are inseparable from the experience of the judge, especially for ethics and morality. It is difficult for AI to accurately predict the outcome of complex cases while taking into account these empirical factors. Therefore, we need to formulate the AI trial system in a human–machine interaction mode, and enable judges to provide real-time feedback on algorithm results.

Contributors

Bin WEI, Kun KUANG, Changlong SUN, and Jun FENG discussed the organization of this paper from different aspects, including the views of both law and computer science. Bin WEI drafted mainly Sections 1, 3, 4, and 10. Kun KUANG drafted mainly Sections 6 and 7. Changlong SUN drafted mainly Sections 2 and 9 and provided judicial big data and technical models for experiments in Section 8. Jun FENG drafted mainly Section 5 and conducted the experiments in Section 8. Fei WU, Xinli ZHU, and Jianghong ZHOU guided the research. All authors revised and finalized the paper.

Acknowledgements

We thank all members of the FITS project team, especially the natural language processing team. In particular, we would like to thank Xiaozhong LIU, Lin YUAN, Huasha ZHAO, Yi YANG, Tianyi WANG, Xinyu DUAN, Qiong ZHANG, Xiaojing LIU, and Feiyu GAO.

Compliance with ethics guidelines

Bin WEI, Kun KUANG, Changlong SUN, Jun FENG, Yating ZHANG, Xinli ZHU, Jianghong ZHOU, Yinsheng ZHAI, and Fei WU declare that they have no conflict of interest.

References

Aletras N, Tsarapatsanis D, Preoţiuc-Pietro D, et al., 2016. Predicting judicial decisions of the European court of human rights: a natural language processing perspective. *PeerJ Comput Sci*, 2:e93. https://doi.org/10.7717/peerj-cs.93

Arditi D, Oksay FE, Tokdemir OB, 1998. Predicting the outcome of construction litigation using neural networks. *Comput-Aided Civ Infrastruct Eng*, 13(2):75-81. https://doi.org/10.1111/0885-9507.00087

Ashley KD, Brüninghaus S, 2009. Automatically classifying case texts and predicting outcomes. *Artif Intell Law*, 17(2):125-165. https://doi.org/10.1007/s10506-009-9077-9

Chao WH, Jiang X, Luo ZC, et al., 2019. Interpretable charge prediction for criminal cases with dynamic rationale attention. *J Artif Intell Res*, 66:743-764. https://doi.org/10.1613/jair.1.11377

Dahbur K, Muscarello T, 2003. Classification system for serial criminal patterns. *Artif Intell Law*, 11(4):251-269. https://doi.org/10.1023/B:ARTI.0000045994.96685.21

Duan XY, Zhang YT, Yuan L, et al., 2019. Legal summarization for multi-role debate dialogue via controversy focus mining and multi-task learning. Proc 28th ACM Int Conf on Information and Knowledge Management, p.1361-1370. https://doi.org/10.1145/3357384.3357940

Elnaggar A, Otto R, Matthes F, 2018. Deep learning for named-entity linking with transfer learning for legal documents. Proc Artificial Intelligence and Cloud Computing Conf, p.23-28. https://doi.org/10.1145/3299819.3299846

Gerani S, Mehdad Y, Carenini G, et al., 2014. Abstractive summarization of product reviews using discourse structure. Proc Conf on Empirical Methods in Natural Language Processing, p.1602-1613.

Goo CW, Chen YN, 2018. Abstractive dialogue summarization with sentence-gated modeling optimized by dialogue acts. IEEE Spoken Language Technology Workshop, p.735-742. https://doi.org/10.1109/SLT.2018.8639531

Graves A, Schmidhuber J, 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neur Netw*, 18(5-6):602-610. https://doi.org/10.1016/j.neunet.2005.06.042

Hochreiter S, Schmidhuber J, 1997. Long short-term memory. *Neur Comput*, 9(8):1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hu YK, Luo ZC, Chao WH, 2020. Identifying principals and accessories in a complex case based on the comprehension of fact description. Proc 58th Annual Meeting of the Association for Computational Linguistics, p.4265-4269. https://doi.org/10.18653/v1/2020.acl-main.393

Imbens GW, Rubin DB, 2015. Causal Inference for Statistics, Social, and Biomedical Sciences: an Introduction. Cambridge University Press, New York, USA.

Jackson P, Al-Kofahi K, Tyrrell A, et al., 2003. Information extraction from case law and retrieval of prior cases. *Artif Intell*, 150(1-2):239-290. https://doi.org/10.1016/S0004-3702(03)00106-1

Ji CZ, Zhou X, Zhang YT, et al., 2020. Cross copy network for dialogue generation. Proc Conf on Empirical Methods in Natural Language Processing, p.1900-1910. https://doi.org/10.18653/v1/2020.emnlp-main.149

Kanapala A, Jannu S, Pamula R, 2019. Passage-based text summarization for legal information retrieval. *Arab J Sci Eng*, 44(11):9159-9169. https://doi.org/10.1007/s13369-019-03998-1

Katz DM, Bommarito MJII, Blackman J, 2017. A general approach for predicting the behavior of the supreme court of the United States. *PLOS ONE*, 12(4):e0174698. https://doi.org/10.1371/journal.pone.0174698

Klement EP, Mesiar R, Pap E, 2000. Triangular Norms. Springer, Dordrecht, the Netherlands. https://doi.org/10.1007/978-94-015-9540-7

Kuang K, Li L, Geng Z, et al., 2020. Causal inference. *Engineering*, 6(3):253-263. https://doi.org/10.1016/j.eng.2019.08.016

Lafferty JD, McCallum A, Pereira FCN, 2001. Conditional random fields: probabilistic models for segmenting and labeling sequence data. Proc 18th Int Conf on Machine Learning, p.282-289.

Lample G, Ballesteros M, Subramanian S, et al., 2016. Neural architectures for named entity recognition. https://arxiv.org/abs/1603.01360

Li T, Gupta V, Mehta M, et al., 2019. A logic-driven framework for consistency of neural models. Proc Conf on Empirical Methods in Natural Language Processing and the 9th Int Joint Conf on Natural Language Processing, p.3924-3935. https://doi.org/10.18653/v1/D19-1405

Li T, Jawale PA, Palmer M, et al., 2020. Structured tuning for semantic role labeling. Proc 58th Annual Meeting of the Association for Computational Linguistics, p.8402-8412.

Liu CL, Chen KC, 2019. Extracting the gist of Chinese judgments of the supreme court. Proc 17th Int Conf on Artificial Intelligence and Law, p.73-82. https://doi.org/10.1145/3322640.3326715

Liu CY, Wang P, Xu J, et al., 2019. Automatic dialogue summary generation for customer service. Proc 25th ACM SIGKDD Int Conf on Knowledge Discovery & Data Mining, p.1957-1965. https://doi.org/10.1145/3292500.3330683

Liu XJ, Gao FY, Zhang Q, et al., 2018. Graph convolution for multimodal information extraction from visually rich documents. Proc NAACL-HLT 2019, p.32-39.

Luo BF, Feng YS, Xu JB, et al., 2017. Learning to predict charges for criminal cases with legal basis. Proc Conf on Empirical Methods in Natural Language Processing, p.2727-2736.

Medvedeva M, Vols M, Wieling M, 2020. Using machine learning to predict decisions of the European court of human rights. *Artif Intell Law*, 28(2):237-266. https://doi.org/10.1007/s10506-019-09255-y

Možina M, Žabkar J, Bench-Capon T, et al., 2005. Argument based machine learning applied to law. *Artif Intell Law*, 13(1):53-73. https://doi.org/10.1007/s10506-006-9002-4

Pearl J, 2009. Causality: Models, Reasoning, and Inference (2nd Ed.). Cambridge University Press, New York, USA.

Pearl J, Glymour M, Jewell NP, 2016. Causal Inference in Statistics: a Primer. John Wiley & Sons, Chichester, UK.

Sak H, Senior A, Beaufays F, 2014. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. Proc 15th Annual Conf of the Int Speech Communication Association, p.338-342.

Sutton C, McCallum A, 2007. An introduction to conditional random fields for relational learning. In: Getoor L, Taskar B (Eds.), Introduction to Statistical Relational Learning. MIT Press, Cambridge, USA, p.268-373.

Wang TY, Zhang YT, Liu XZ, et al., 2020. Masking orchestration: multi-task pretraining for multi-role dialogue representation learning. Proc 34th AAAI Conf on Artificial Intelligence, p.9217-9224. https://doi.org/10.1609/aaai.v34i05.6459

Wu YQ, Kuang K, Zhang YT, et al., 2020. De-biased court's view generation with causality. Proc Conf on Empirical Methods in Natural Language Processing, p.763-780.

Xiao CJ, Zhong HX, Guo ZP, et al., 2018. CAIL2018: a large-scale legal dataset for judgment prediction. https://arxiv.org/abs/1807.02478

Xie YQ, Xu ZW, Kankanhalli MS, et al., 2019. Embedding symbolic knowledge into deep networks. Proc 33rd Conf on Neural Information Processing Systems, p.4233-4243.

Yang WM, Jia WJ, Zhou XJ, et al., 2019. Legal judgment prediction via multi-perspective bi-feedback network. Proc 28th Int Joint Conf on Artificial Intelligence, p.4085-4091.

Yang ZC, Yang DY, Dyer C, et al., 2016. Hierarchical attention networks for document classification. Proc Conf of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, p.1480-1489. https://doi.org/10.18653/v1/N16-1174

Yang ZL, Salakhutdinov R, Cohen WW, 2017. Transfer learning for sequence tagging with hierarchical recurrent networks. Proc Int Conf on Learning Representations.

Ye H, Jiang X, Luo ZC, et al., 2018. Interpretable charge predictions for criminal cases: learning to generate court views from fact descriptions. Proc Conf of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, p.1854-1864. https://doi.org/10.18653/v1/N18-1168

Zhao HS, Yang Y, Zhang Q, et al., 2018. Improve neural entity recognition via multi-task data selection and constrained decoding. Proc Conf of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, p.346-351. https://doi.org/10.18653/v1/N18-2056

Zhong HX, Guo ZP, Tu CC, et al., 2018. Legal judgment prediction via topological learning. Proc Conf on Empirical Methods in Natural Language Processing, p.3540-3549. https://doi.org/10.18653/v1/D18-1390

Zhong HX, Xiao CJ, Tu CC, et al., 2020a. How does NLP benefit legal system: a summary of legal artificial intelligence. Proc 58th Annual Meeting of the Association for Computational Linguistics, p.5218-5230. https://doi.org/10.18653/v1/2020.acl-main.466

Zhong HX, Wang YZ, Tu CC, et al., 2020b. Iteratively questioning and answering for interpretable legal judgment prediction. Proc AAAI Conf on Artificial Intelligence, p.1250-1257. https://doi.org/10.1609/aaai.v34i01.5479

Zhou X, Zhang YT, Liu XZ, et al., 2019. Legal intelligence for e-commerce: multi-task learning by leveraging multiview dispute representation. Proc 42nd Int ACM SIGIR Conf on Research and Development in Information Retrieval, p.315-324. https://doi.org/10.1145/3331184.3331212