







# Automatic Judgment Prediction via Legal Reading Comprehension

Shangbang Long<sup>1</sup> , Cunchao Tu<sup>2</sup> , Zhiyuan Liu<sup>2</sup>  , and Maosong Sun<sup>2</sup>

<sup>1</sup> Department of Machine Learning, Carnegie Mellon University, Pittsburgh, USA  
longshangbang@cmu.edu

<sup>2</sup> Department of CST, Tsinghua University, Beijing, China  
tucunchao@gmail.com, {liuzy,sms}@tsinghua.edu.cn

**Abstract.** Automatic judgment prediction aims to predict the judicial results based on case materials. It has been studied for several decades mainly by lawyers and judges, considered as a novel and prospective application of artificial intelligence techniques in the legal field. Most existing methods follow the text classification framework, which fails to model the complex interactions among complementary case materials. To address this issue, we formalize the task as **Legal Reading Comprehension** according to the legal scenario. Following the working protocol of human judges, LRC predicts the final judgment results based on three types of information, including *fact description*, plaintiffs' *pleas*, and *law articles*. Moreover, we propose a novel LRC model, **AutoJudge**, which captures the complex semantic interactions among facts, pleas, and laws. In experiments, we construct a real-world civil case dataset for LRC. Experimental results on this dataset demonstrate that our model achieves significant improvement over state-of-the-art models. We have published all source codes of this work on <https://github.com/thunlp/AutoJudge>.

## 1 Introduction

Automatic judgment prediction is to train a machine judge to determine whether a certain *plea* in a given civil case would be supported or rejected. In countries with *civil law system*, e.g. mainland China, such process should be done with reference to related *law articles* and the *fact description*, as is performed by a human judge. The intuition comes from the fact that under *civil law system*, law articles act as principles for juridical judgments. Such techniques would have a wide range of promising applications. On the one hand, legal consulting systems could provide better access to high-quality legal resources in a low-cost way to legal outsiders, who suffer from the complicated terminologies. On the other hand, machine judge assistants for professionals would help improve the efficiency of the judicial system. Besides, automated judgment system can help in improving juridical equality and transparency. From another perspective, there are currently 7 times much more civil cases than criminal cases in mainland

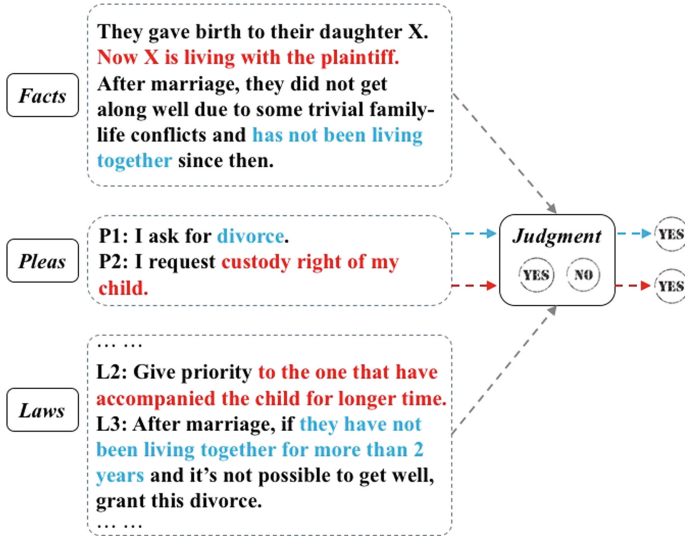


Fig. 1. An example of LRC.

China, with annual rates of increase of 10.8% and 1.6% respectively, making judgment prediction in civil cases a promising application [40].

Previous works [1, 12, 23, 33] formalize judgment prediction as the text classification task, regarding either charge names or binary judgments, i.e., support or reject, as the target classes. These works focus on the situation where only one result is expected, e.g., the US Supreme Court's decisions [12], and the charge name prediction for criminal cases [23]. Despite these recent efforts and their progress, automatic judgment prediction in *civil law system* is still confronted with two main challenges:

**One-to-Many Relation between Case and Plea.** Every single civil case may contain multiple pleas and the result of each plea is co-determined by related law articles and specific aspects of the involved case. For example, in divorce proceedings, judgment of *alienation of mutual affection* is the key factor for *granting divorce* but *custody of children* depends on which side can provide better an environment for children's growth as well as parents' financial condition. Here, different pleas are independent.

**Heterogeneity of Input Triple.** Inputs to a judgment prediction system consist of three heterogeneous yet complementary parts, i.e., *fact description*, *plaintiff's plea*, and *related law articles*. Concatenating them together and treating them simply as a sequence of words as in previous works [1, 12] would cause a great loss of information. This is the same in *question-answering* where the dual inputs, i.e., *query* and *passage*, should be modeled separately.

Despite the introduction of the neural networks that can learn better semantic representations of input text, it remains unsolved to incorporate proper mech-

anisms to integrate the complementary triple of *pleas*, *fact descriptions*, and *law articles* together.

Inspired by recent advances in question answering (QA) based reading comprehension (RC) [7, 28, 29, 37], we propose the **Legal Reading Comprehension (LRC)** framework for automatic judgment prediction. LRC incorporates the reading mechanism for better modeling of the complementary inputs above-mentioned, as is done by human judges when referring to legal materials in search of supporting *law articles*. Reading mechanism, by simulating how human connects and integrates multiple text, has proven an effective module in RC tasks. We argue that applying the reading mechanism in a proper way among the triplets can obtain a better understanding and more informative representation of the original text, and further improve performance. To instantiate the framework, we propose an end-to-end neural network model named **AutoJudge**.

For experiments, we train and evaluate our models in the civil law system of mainland China. We collect and construct a large-scale real-world data set of 100,000 case documents that the Supreme People’s Court of People’s Republic of China has made publicly available. *Fact description*, *pleas*, and *results* can be extracted easily from these case documents with regular expressions, since the original documents have special typographical characteristics indicating the discourse structure. We also take into account law articles and their corresponding juridical interpretations. We also implement and evaluate previous methods on our dataset, which prove to be strong baselines.

Our experiment results show significant improvements over previous methods. Further experiments demonstrate that our model also achieves considerable improvement over other off-the-shelf state-of-the-art models under classification and question answering framework respectively. Ablation tests carried out by taking off some components of our model further prove its robustness and effectiveness.

To sum up, our contributions are as follows:

- (1) We introduce reading mechanism and re-formalize judgment prediction as Legal Reading Comprehension to better model the complementary inputs.
- (2) We construct a real-world dataset for experiments, and plan to publish it for further research.
- (3) Besides baselines from previous works, we also carry out comprehensive experiments comparing different existing deep neural network methods on our dataset. Supported by these experiments, improvements achieved by LRC prove to be robust.

## 2 Related Work

### 2.1 Judgment Prediction

Automatic judgment prediction has been studied for decades. At the very first stage of judgment prediction studies, researchers focus on mathematical and

statistical analysis of existing cases, without any conclusions or methodologies on how to predict them [13, 16, 17, 27, 31, 35].

Recent attempts consider judgment prediction under the text classification framework. Most of these works extract efficient features from text (e.g., N-grams) [1, 18, 21, 22, 33, 39] or case profiles (e.g., dates, terms, locations and types) [12]. All these methods require a large amount of human effort to design features or annotate cases. Besides, they also suffer from generalization issue when applied to other scenarios.

Motivated by the successful application of deep neural networks, Luo et al. [23] introduce an attention-based neural model to predict charges of criminal cases, and verify the effectiveness of taking law articles into consideration. Nevertheless, they still fall into the text classification framework and lack the ability to handle multiple inputs with more complicated structures.

## 2.2 Text Classification

As the basis of previous judgment prediction works, typical text classification task takes a single text content as input and predicts the category it belongs to. Recent works usually employ neural networks to model the internal structure of a single input [2, 14, 34, 38].

There also exists another thread of text classification called entailment prediction. Methods proposed in [10, 26] are intended for complementary inputs, but the mechanisms can be considered as a simplified version of reading comprehension.

## 2.3 Reading Comprehension

Reading comprehension is a relevant task to model heterogeneous and complementary inputs, where an *answer* is predicted given two channels of inputs, i.e. a textual *passage* and a *query*. Considerable progress has been made [7, 8, 37]. These models employ various attention mechanism to model the interaction between *passage* and *query*. Inspired by the advantage of reading comprehension models on modeling multiple inputs, we apply this idea into the legal area and propose legal reading comprehension for judgment prediction.

# 3 Legal Reading Comprehension

## 3.1 Conventional Reading Comprehension

Conventional reading comprehension [9, 11, 28, 29] usually considers reading comprehension as predicting the *answer* given a *passage* and a *query*, where the *answer* could be a single word, a text span of the original *passage*, chosen from answer candidates, or generated by human annotators.

Generally, an instance in RC is represented as a triple  $\langle p, q, a \rangle$ , where  $p$ ,  $q$  and  $a$  correspond to *passage*, *query* and *answer* respectively. Given a triple  $\langle p, q, a \rangle$ ,

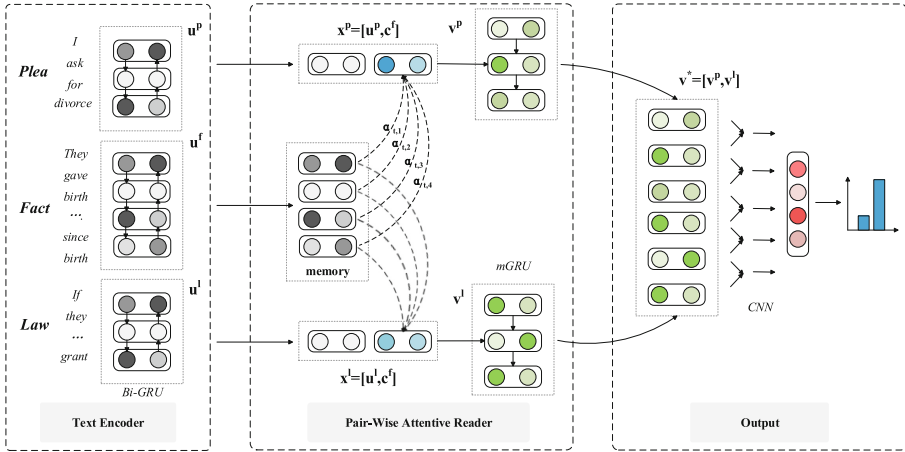


Fig. 2. An overview of AutoJudge.

RC takes the pair  $\langle p, q \rangle$  as the input and employs attention-based neural models to construct an efficient representation. Afterwards, the representation is fed into the output layer to select or generate an *answer*.

### 3.2 Legal Reading Comprehension

Existing works usually formalize judgment prediction as a text classification task and focus on extracting well-designed features of specific cases. Such simplification ignores that the judgment of a case is determined by its fact description and multiple pleas. Moreover, the final judgment should act up to the legal provisions, especially in civil law systems. Therefore, how to integrate the information (i.e., *fact descriptions*, *pleas*, and *law articles*) in a reasonable way is critical for judgment prediction.

Inspired by the successful application of RC, we propose a framework of **Legal Reading Comprehension (LRC)** for judgment prediction in the legal area. As illustrated in Fig. 1, for each *plea* in a given case, the prediction of judgment *result* is made based the *fact description* and the potentially relevant *law articles*.

In a nutshell, LRC can be formalized as the following quadruplet task:

$$\langle f, p, l, r \rangle, \tag{1}$$

where  $f$  is the *fact description*,  $p$  is the *plea*,  $l$  is the *law articles* and  $r$  is the *result*. Given  $\langle f, p, l \rangle$ , LRC aims to predict the judgment result as

$$r = \arg \max_{r \in \{support, reject\}} P(r|f, p, l). \tag{2}$$

The probability is calculated with respect to the interaction among the triple  $\langle f, p, l \rangle$ , which will draw on the experience of the interaction between  $\langle passage, question \rangle$  pairs in RC.

To summarize, LRC is innovative in the following aspects:

- (1) While previous works fit the problem into text classification framework, LRC re-formalizes the way to approach such problems. This new framework provides the ability to deal with the heterogeneity of the complementary inputs.
- (2) Rather than employing conventional RC models to handle pair-wise text information in the legal area, LRC takes the critical law articles into consideration and models the facts, pleas, and law articles jointly for judgment prediction, which is more suitable to simulate the human mode of dealing with cases.

## 4 Methods

We propose a novel judgment prediction model **AutoJudge** to instantiate the LRC framework. As shown in Fig. 2, AutoJudge consists of three flexible modules, including a text encoder, a pair-wise attentive reader, and an output module.

In the following parts, we give a detailed introduction to these three modules.

### 4.1 Text Encoder

As illustrated in Fig. 2, **Text Encoder** aims to encode the word sequences of inputs into continuous representation sequences.

Formally, consider a fact description  $f = \{w_t^f\}_{t=1}^m$ , a plea  $p = \{w_t^p\}_{t=1}^n$ , and the relevant law articles  $l = \{w_t^l\}_{t=1}^k$ , where  $w_t$  denotes the  $t$ -th word in the sequence and  $m, n, k$  are the lengths of word sequences  $f, p, l$  respectively.

First, we convert the words to their respective word embeddings to obtain  $\mathbf{f} = \{\mathbf{w}_t^f\}_{t=1}^m$ ,  $\mathbf{p} = \{\mathbf{w}_t^p\}_{t=1}^n$  and  $\mathbf{l} = \{\mathbf{w}_t^l\}_{t=1}^k$ , where  $\mathbf{w} \in R^d$ . Afterwards, we employ bi-directional GRU [3,5,6] to produce the encoded representation  $\mathbf{u}$  of all words as follows:

$$\begin{aligned} \mathbf{u}_t^f &= \text{BiGRU}_F(\mathbf{u}_{t-1}^f, \mathbf{w}_t^f), \\ \mathbf{u}_t^p &= \text{BiGRU}_P(\mathbf{u}_{t-1}^p, \mathbf{w}_t^p), \\ \mathbf{u}_t^l &= \text{BiGRU}_L(\mathbf{u}_{t-1}^l, \mathbf{w}_t^l). \end{aligned} \quad (3)$$

Note that, we adopt different bi-directional GRUs to encode fact descriptions, pleas, and law articles respectively (denoted as  $\text{BiGRU}_F$ ,  $\text{BiGRU}_P$ , and  $\text{BiGRU}_L$ ). With these text encoders,  $f$ ,  $p$ , and  $l$  are converting into  $\mathbf{u}^f = \{\mathbf{u}_t^f\}_{t=1}^m$ ,  $\mathbf{u}^p = \{\mathbf{u}_t^p\}_{t=1}^n$ , and  $\mathbf{u}^l = \{\mathbf{u}_t^l\}_{t=1}^k$ .

### 4.2 Pair-Wise Attentive Reader

How to model the interactions among the input text is the most important problem in reading comprehension. In AutoJudge, we employ a pair-wise attentive reader to process  $\langle \mathbf{u}^f, \mathbf{u}^p \rangle$  and  $\langle \mathbf{u}^f, \mathbf{u}^l \rangle$  respectively. More specifically, we

propose to use pair-wise mutual attention mechanism to capture the complex semantic interaction between text pairs, as well as increasing the interpretability of AutoJudge.

**Pair-Wise Mutual Attention.** For each input pair  $\langle \mathbf{u}^f, \mathbf{u}^p \rangle$  or  $\langle \mathbf{u}^f, \mathbf{u}^l \rangle$ , we employ pair-wise mutual attention to select relevant information from fact descriptions  $\mathbf{u}^f$  and produce more informative representation sequences.

As a variant of the original attention mechanism [3], we design the pair-wise mutual attention unit as a GRU with internal memories denoted as **mGRU**.

Taking the representation sequence pair  $\langle \mathbf{u}^f, \mathbf{u}^p \rangle$  for instance, mGRU stores the fact sequence  $\mathbf{u}^f$  into its memories. For each timestamp  $t \in [1, n]$ , it selects relevant fact information  $\mathbf{c}_t^f$  from the memories as follows,

$$\mathbf{c}_t^f = \sum_{i=1}^m \alpha_{t,i} \mathbf{u}_i^f. \quad (4)$$

Here, the weight  $\alpha_{t,i}$  is the softmax value as

$$\alpha_{t,i} = \frac{\exp(a_{t,i})}{\sum_{j=1}^m \exp(a_{t,j})}. \quad (5)$$

Note that,  $a_{t,j}$  represents the relevance between  $\mathbf{u}_t^p$  and  $\mathbf{u}_j^f$ . It is calculated as follows,

$$a_{t,j} = \mathbf{V}^T \tanh(\mathbf{W}^f \mathbf{u}_j^f + \mathbf{W}^p \mathbf{u}_t^p + \mathbf{U}^p \mathbf{v}_{t-1}^p). \quad (6)$$

Here,  $\mathbf{v}_{t-1}^p$  is the last hidden state in the GRU, which will be introduced in the following part.  $\mathbf{V}$  is a weight vector, and  $\mathbf{W}^f$ ,  $\mathbf{W}^p$ ,  $\mathbf{U}^p$  are attention metrics of our proposed pair-wise attention mechanism.

**Reading Mechanism.** With the relevant fact information  $\mathbf{c}_t^f$  and  $\mathbf{u}_t^p$ , we get the  $t$ -th input of mGRU as

$$\mathbf{x}_t^p = \mathbf{u}_t^p \oplus \mathbf{c}_t^f, \quad (7)$$

where  $\oplus$  indicates the concatenation operation.

Then, we feed  $\mathbf{x}_t^p$  into GRU to get more informative representation sequence  $\mathbf{v}^p = \{\mathbf{v}_t^p\}_{t=1}^n$  as follows,

$$\mathbf{v}_t^p = \text{GRU}(\mathbf{v}_{t-1}^p, \mathbf{x}_t^p). \quad (8)$$

For the input pair  $\langle \mathbf{u}^f, \mathbf{u}^l \rangle$ , we can get  $\mathbf{v}^l = \{\mathbf{v}_t^l\}_{t=1}^k$  in the same way. Therefore, we omit the implementation details Here.

Similar structures with attention mechanism are also applied in [3,30,36,37] to obtain mutually aware representations in reading comprehension models, which significantly improve the performance of this task.

### 4.3 Output Layer

Using text encoder and pair-wise attentive reader, the initial input triple  $\langle f, p, l \rangle$  has been converted into two sequences, i.e.,  $\mathbf{v}^p = \{\mathbf{v}_t^p\}_{t=1}^n$  and  $\mathbf{v}^l = \{\mathbf{v}_t^l\}_{t=1}^k$ ,

where  $\mathbf{v}_t^l$  is defined similarly to  $\mathbf{v}_t^p$ . These sequences reserve complex semantic information about the pleas and law articles, and filter out irrelevant information in fact descriptions.

With these two sequences, we concatenate  $\mathbf{v}^p$  and  $\mathbf{v}^l$  along the *sequence length* dimension to generate the sequence  $\mathbf{v}^* = \{\mathbf{v}_t\}_{t=1}^{n+k}$ . Since we have employed several GRU layers to encode the sequential inputs, another recurrent layer may be redundant. Therefore, we utilize a 1-layer CNN [14] to capture the local structure and generate the representation vector for the final prediction.

Assuming  $y \in [0, 1]$  is the predicted probability that the plea in the case sample would be supported and  $r \in \{0, 1\}$  is the gold standard, AutoJudge aims to minimize the cross-entropy as follows,

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N [r_i \ln y_i + (1 - r_i) \ln(1 - y_i)], \quad (9)$$

where  $N$  is the number of training data. As all the calculation in our model is differentiable, we employ Adam [15] for optimization.

## 5 Experiments

To evaluate the proposed LRC framework and the AutoJudge model, we carry out a series of experiments on the divorce proceedings, a typical yet complex field of civil cases. Divorce proceedings often come with several kinds of *pleas*, e.g. *seeking divorce*, *custody of children*, *compensation*, and *maintenance*, which focuses on different aspects and thus makes it a challenge for judgment prediction.

### 5.1 Dataset Construction for Evaluation

**Data Collection.** Since none of the datasets from previous works have been published, we decide to build a new one. We randomly collect 100,000 cases from China Judgments Online<sup>1</sup>, among which 80,000 cases are for training, 10,000 each for validation and testing. Among the original cases, 51% are granted *divorce* and others not. There are 185,723 valid *pleas* in total, with 52% supported and 48% rejected. Note that, if the *divorce plea* in a case is not granted, the other *pleas* of this case will not be considered by the judge. Case materials are all natural language sentences, with averagely 100.08 tokens per *fact description* and 12.88 per *plea*. There are 62 relevant *law articles* in total, each with 26.19 tokens averagely. Note that the case documents include special typographical signals, making it easy to extract labeled data with regular expression.

<sup>1</sup> <http://wenshu.court.gov.cn>.



**Data Pre-Processing.** We apply some rules with legal prior to preprocess the dataset according to previous works [4, 19, 20], which have proved effective in our experiments.

**Name Replacement**<sup>2</sup>: All names in case documents are replaced with marks indicating their roles, instead of simply anonymizing them, e.g.  $\langle Plaintiff \rangle$ ,  $\langle Defendant \rangle$ ,  $\langle Daughter_x \rangle$  and so on. Since “*all are equal before the law*”<sup>3</sup>, names should make no more difference than what role they take.

**Law Article Filtration:** Since most accessible divorce proceeding documents do not contain ground-truth *fine-grained* articles<sup>4</sup>, we use an unsupervised method instead. First, we extract all the articles from the law text with regular expression. Afterwards, we select the most relevant 10 articles according to the fact descriptions as follows. We obtain sentence representation with CBOV [24, 25] weighted by inverse document frequency, and calculate cosine distance between cases and law articles. Word embeddings are pre-trained with Chinese Wikipedia pages<sup>5</sup>. As the final step, we extract top 5 relevant articles for each sample respectively from the main marriage law articles and their interpretations, which are equally important. We manually check the extracted articles for 100 cases to ensure that the extraction quality is fairly good and acceptable.

The filtration process is automatic and fully unsupervised since the original documents have no ground-truth labels for fine-grained law articles, and coarse-grained law-articles only provide limited information. We also experiment with the ground-truth articles, but only a small fraction of them has fine-grained ones, and they are usually not available in real-world scenarios.

## 5.2 Implementation Details

We employ Jieba<sup>6</sup> for Chinese word segmentation and keep the top 20,000 frequent words. The word embedding size is set to 128 and the other low-frequency words are replaced with the mark  $\langle UNK \rangle$ . The hidden size of GRU is set to 128 for each direction in Bi-GRU. In the pair-wise attentive reader, the hidden state is set to 256 for mGRU. In the CNN layer, filter windows are set to 1, 3, 4, and 5 with each filter containing 200 feature maps. We add a dropout layer [32] after the CNN layer with a dropout rate of 0.5. We use Adam [15] for training and set learning rate to  $10^{-4}$ ,  $\beta_1$  to 0.9,  $\beta_2$  to 0.999,  $\epsilon$  to  $10^{-8}$ . We employ *precision*, *recall*, *F1* and *accuracy* for evaluation metrics. We repeat all the experiments for 10 times, and report the average results.

<sup>2</sup> We use regular expressions to extract *names* and *roles* from the formatted case header.

<sup>3</sup> Constitution of the People’s Republic of China.

<sup>4</sup> *Fine-grained* articles are in the *Judicial Interpretations*, giving detailed explanation, while *the Marriage Law* only covers some basic principles.

<sup>5</sup> <https://dumps.wikimedia.org/zhwiki/>.

<sup>6</sup> <https://github.com/fxsjy/jieba>.

**Table 1.** Experimental results(%). Precision/Recall/F1 are reported for positive samples and calculated as the mean score over 10-time experiments. Acc is defined as the proportion of test samples classified correctly, equal to micro-precision. MaxFreq refers to always predicting the most frequent label, i.e. *support* in our dataset. \* indicates methods proposed in previous works.

Models	Precision	Recall	F1	Accuracy
MaxFreq	52.2	100	68.6	52.2
SVM*	57.8	53.5	55.6	55.5
CNN	76.1	81.9	79.0	77.6
CNN+law	74.4	79.4	77.0	76.0
GRU	79.2	72.9	76.1	76.6
GRU+law	78.2	68.2	72.8	74.4
GRU+Attention*	79.1	80.7	80.0	79.1
AoA	79.3	78.9	79.2	78.3
AoA+law	79.0	79.2	79.1	78.3
r-net	79.5	78.7	79.2	78.4
r-net+law	79.3	78.8	79.0	78.3
<b>AutoJudge</b>	<b>80.4</b>	<b>86.6</b>	<b>83.4</b>	<b>82.2</b>

### 5.3 Baselines

For comparison, we adopt and re-implement three kinds of baselines as follows:

**Lexical Features SVM.** We implement an SVM with lexical features in accordance with previous works [1, 18, 21, 22, 33] and select the best feature set on the development set.

**Neural Text Classification Models.** We implement and fine-tune a series of neural text classifiers, including attention-based method [23] and other methods we deem important. CNN [14] and GRU [5, 38] take as input the concatenation of *fact description* and *plea*. Similarly, *CNN/GRU+law* refers to using the concatenation of *fact description*, *plea* and *law articles* as inputs.

**RC Models.** We implement and train some off-the-shelf RC models, including r-net [37] and AoA [7], which are the leading models on SQuAD leaderboard. In our initial experiments, these models take *fact description* as *passage* and *plea* as *query*. Further, *Law articles* are added to the *fact description* as a part of the reading materials, which is a simple way to consider them as well.

From Table 1, we have the following observations:

- (1) AutoJudge consistently and significantly outperforms all the baselines, including RC models and other neural text classification models, which shows the effectiveness and robustness of our model.

- (2) RC models achieve better performance than most text classification models (excluding *GRU+Attention*), which indicates that reading mechanism is a better way to integrate information from heterogeneous yet complementary inputs. On the contrary, simply adding *law articles* as a part of the reading materials makes no difference in performance. Note that, *GRU+Attention* employ similar attention mechanism as RC does and takes additional law articles into consideration, thus achieves comparable performance with RC models.
- (3) Comparing with conventional RC models, AutoJudge achieves significant improvement with the consideration of additional law articles. It reflects the difference between LRC and conventional RC models. We re-formalize LRC in legal area to incorporate law articles via the reading mechanism, which can enhance judgment prediction. Moreover, CNN/GRU+law decrease the performance by simply concatenating original text with law articles, while *GRU+Attention/AutoJudge* increase the performance by integrating law articles with attention mechanism. It shows the importance and rationality of using attention mechanism to capture the interaction between multiple inputs.

The experiments support our hypothesis as proposed in the Introduction part that in civil cases, it's important to model the interactions among case materials. Reading mechanism can well perform the matching among them.

#### 5.4 Ablation Test

AutoJudge is characterized by the incorporation of pair-wise attentive reader, law articles, and a CNN output layer, as well as some pre-processing with legal prior. We design ablation tests respectively to evaluate the effectiveness of these modules. When taken off the attention mechanism, AutoJudge degrades into a GRU on which a CNN is stacked. When taken off law articles, the CNN output layer only takes  $\{v_t^P\}_{t=1}^{L_P}$  as input. Besides, our model is tested respectively without name-replacement or unsupervised selection of law articles (i.e. passing the whole law text). As mentioned above, we system use law articles extracted with unsupervised method, so we also experiment with ground-truth law articles.

Results are shown in Table 2. We can infer that:

- (1) The performance drops significantly after removing the attention layer or excluding the law articles, which is consistent with the comparison between AutoJudge and baselines. The result verifies that both the reading mechanism and incorporation of law articles are important and effective.
- (2) After replacing CNN with an LSTM layer, performance drops as much as 4.4% in accuracy and 5.7% in F1 score. The reason may be the redundancy of RNNs. AutoJudge has employed several GRU layers to encode text sequences. Another RNN layer may be useless to capture sequential dependencies, while CNN can catch the local structure in convolution windows.

**Table 2.** Experimental results of ablation tests (%).

Models	F1	Accuracy
AutoJudge	83.4	82.2
w/o reading mechanism	78.9(↓ 4.5)	78.2(↓ 4.0)
w/o law articles	79.6(↓ 3.8)	78.4(↓ 3.8)
CNN→LSTM	77.6(↓ 5.8)	77.7(↓ 4.5)
w/o Pre-Processing	81.1(↓ 2.3)	80.3(↓ 1.9)
w/o law article selection	80.6(↓ 2.8)	80.5(↓ 1.7)
with GT law articles	<b>85.1</b> (↑ 1.7)	<b>84.1</b> (↑ 1.9)

- (3) Motivated by existing rule-based works, we conduct data pre-processing on cases, including name replacement and law article filtration. If we remove the pre-processing operations, the performance drops considerably. It demonstrates that applying the prior knowledge in legal filed would benefit the understanding of legal cases.

**Performance over Law Articles.** It’s intuitive that the quality of the retrieved *law articles* would affect the final performance. As is shown in Table 2, feeding the whole law text without filtration results in worse performance. However, when we train and evaluate our model with ground truth articles, the performance is boosted by nearly 2% in both F1 and Acc. The performance improvement is quite limited compared to that in previous work [23] for the following reasons: (1) As mentioned above, most case documents only contain coarse-grained articles, and only a small number of them contain fine-grained ones, which has limited information in themselves. (2) Unlike in criminal cases where the application of an article indicates the corresponding crime, *law articles* in civil cases work as reference, and can be applied in both the cases of *supports* and *rejects*. As law articles cut both ways for the judgment result, this is one of the characteristics that distinguishes civil cases from criminal ones. We also need to remember that, the performance of 84.1% in accuracy or 85.1% in F1 score is unattainable in real-world setting for automatic prediction where ground-truth articles are not available.

**Reading Weighs More Than Correct Law Articles.** In the area of civil cases, the understanding of the case materials and how they interact is a critical factor. The inclusion of law articles is not enough. As is shown in Table 2, compared to feeding the model with an un-selected set of law articles, taking away the reading mechanism results in greater performance drop<sup>7</sup>. Therefore, the ability to read, understand and select relevant information from the complex multi-sourced case materials is necessary. It’s even more important in real world since we don’t have access to ground-truth law articles to make predictions.

<sup>7</sup> 3.9% vs. 1.7% in Acc, and 4.4% vs. 2.8% in F1.

## 5.5 Case Study

**Visualization of Positive Samples.** We visualize the heat maps of attention results<sup>8</sup>. As shown in Fig. 3, deeper background color represents larger attention score.

The attention score is calculated with Eq. (5). We take the average of the resulting  $n \times m$  attention matrix over the time dimension to obtain attention values for each word.

The visualization demonstrates that the attention mechanism can capture relevant patterns and semantics in accordance with different *pleas* in different *cases*.

**Failure Analysis.** As for the failed samples, the most common reason comes from the anonymity issue, which is also shown in Fig. 3. As mentioned above, we conduct name replacement. However, some critical elements are also anonymized by the government, due to the privacy issue. These elements are sometimes important to judgment prediction. For example, determination of the key factor *long-time separation* is relevant to the explicit dates, which are anonymized.

(a) Case 1: Plea 1 - Seeking Divorce

法院查明原被告经人介绍了解后于x年x月x日举行结婚仪式，x年x月x日办理结婚登记手续，x年x月x日生育女儿张某某，现女儿张某某跟随原告生活，婚后双方因家庭琐事产生矛盾纠纷后，双方分居至今。

The Court confirmed that the plaintiff and the defendant met on some day via acquaintance and registered marriage on some day. They gave birth to their daughter X. Now X is living with the plaintiff. After marriage, plaintiff and defendant did not get along well due to some **trivial family-life conflicts** and has **not been living together since then**.

(c) Case 1: Plea 2 - The plaintiff asks for the right of child custody .

法院查明原被告经人介绍了解后于x年x月x日举行结婚仪式，x年x月x日办理结婚登记手续，x年x月x日生育女儿张某某，现女儿张某某跟随原告生活，婚后双方因家庭琐事产生矛盾纠纷后，双方分居至今。

(b) Case 2: Plea 1 - Seeking Divorce

法院查明，原告与被告经人介绍相识后，于x年x月x日于香港登记结婚，x年x月x日生育某乙。x年x月x日，原告以夫妻感情不和为由，向本院提起离婚诉讼，被本院判决不准予离婚。现原被告均同意离婚。

The Court confirmed that the plaintiff and the defendant met on some day via acquaintance and get married in Hong Kong on some day. They gave birth to their child X. In a previous divorce proceeding, the plaintiff claimed that they didn't get along well but was **rejected** divorce. Now **both** parties **agree to get divorced**.

The Court confirmed that the plaintiff and the defendant met on some day via acquaintance and registered marriage on some day. They gave birth to their daughter X. Now X is **living with the plaintiff**. After marriage, plaintiff and defendant did not get along well due to some **trivial family-life conflicts** and has **not been living together since then**.

**Fig. 3.** Visualization of attention mechanism.

## 6 Conclusion

In this paper, we explore the task of predicting judgments of civil cases. Comparing with conventional text classification framework, we propose **Legal Reading Comprehension** framework to handle multiple and complex textual inputs. Moreover, we present a novel neural model, **AutoJudge**, to incorporate law articles for judgment prediction. In experiments, we compare our model on divorce proceedings with various state-of-the-art baselines of various frameworks. Experimental results show that our model achieves considerable improvement than all the baselines. Besides, visualization results also demonstrate the effectiveness and interpretability of our proposed model.

<sup>8</sup> Examples given here are all drawn from the test set whose predictions match the real judgment.

In the future, we can explore the following directions: (1) Limited by the datasets, we can only verify our proposed model on divorce proceedings. A more general and larger dataset will benefit the research on judgment prediction. (2) Judicial decisions in some civil cases are not always binary, but more diverse and flexible ones, e.g. compensation amount. Thus, it is critical for judgment prediction to manage various judgment forms.

**Acknowledgments.** This work is supported by the National Key Research and Development Program of China (No. 2018YFC0831900).

## References

1. Aletras, N., Tsarapatsanis, D., Preotiuc-Pietro, D., Lampos, V.: Predicting judicial decisions of the European court of human rights: a natural language processing perspective. *PeerJ Comput. Sci.* **2**, e93 (2016)
2. Baharudin, B., Lee, L.H., Khan, K.: A review of machine learning algorithms for text-documents classification. *JAIT* **1**(1), 4–20 (2010)
3. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: *Proceedings of ICLR* (2015)
4. Bian, G.W., Shun-yuan, T.: Integrating query translation and text classification in a cross-language patent access system. In: *Proceedings of NTCIR-7 Workshop Meeting*, pp. 252–261 (2005)
5. Cho, K., et al.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. *Computer Science* (2014)
6. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. In: *Proceedings of NIPS* (2014)
7. Cui, Y., Chen, Z., Wei, S., Wang, S., Liu, T., Hu, G.: Attention-over-attention neural networks for reading comprehension. In: *Proceedings of ACL* (2017)
8. Dhingra, B., Liu, H., Yang, Z., Cohen, W.W., Salakhutdinov, R.: Gated-attention readers for text comprehension. In: *Proceedings of ACL* (2017)
9. He, W., et al.: Dureader: a chinese machine reading comprehension dataset from real-world applications. *arXiv preprint [arXiv:1711.05073](https://arxiv.org/abs/1711.05073)* (2017)
10. Hu, B., Lu, Z., Li, H., Chen, Q.: Convolutional neural network architectures for matching natural language sentences. In: *Proceedings of NIPS*, pp. 2042–2050 (2014)
11. Joshi, M., Choi, E., Weld, D.S., Zettlemoyer, L.: Triviaqa: a large scale distantly supervised challenge dataset for reading comprehension. In: *Proceedings of ACL* (2017)
12. Katz, D.M., Bommarito II, M.J., Blackman, J.: A general approach for predicting the behavior of the supreme court of the united states. *PLoS One* **12**(4), e0174698 (2017)
13. Keown, R.: Mathematical models for legal prediction. *Computer/LJ* **2**, 829 (1980)
14. Kim, Y.: Convolutional neural networks for sentence classification. In: *Proceedings of EMNLP* (2014)
15. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. In: *Proceedings of ICLR* (2015)
16. Kort, F.: Predicting supreme court decisions mathematically: a quantitative analysis of the "right to counsel" cases. *Am. Polit. Sci. Rev.* **51**(1), 1–12 (1957)
17. Lauderdale, B.E., Clark, T.S.: The supreme court's many median justices. *Am. Polit. Sci. Rev.* **106**(4), 847–866 (2012)

18. Lin, W.C., Kuo, T.T., Chang, T.J., Yen, C.A., Chen, C.J., Lin, S.D.: Exploiting machine learning models for Chinese legal documents labeling, case classification, and sentencing prediction. In: *Proceedings of ROCLING*, p. 140 (2012)
19. Liu, C.L., Chang, C.T., Ho, J.H.: Case instance generation and refinement for case-based criminal summary judgments in Chinese. *JISE* (2004)
20. Liu, C.L., Ho, J.H., Ho, J.H.: Classification and clustering for case-based criminal summary judgments. In: *Proceedings of the International Conference on Artificial Intelligence and Law*, pp. 252–261 (2003)
21. Liu, C.L., Hsieh, C.D.: Exploring phrase-based classification of judicial documents for criminal charges in Chinese. In: *Proceedings of ISMIS*, pp. 681–690 (2006)
22. Liu, Y.H., Chen, Y.L.: A two-phase sentiment analysis approach for judgement prediction. *J. Inf. Sci.* **44**(5), 594–607 (2017)
23. Luo, B., Feng, Y., Xu, J., Zhang, X., Zhao, D.: Learning to predict charges for criminal cases with legal basis. In: *Proceedings of EMNLP* (2017)
24. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013)
25. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: *Proceedings of NIPS*, pp. 3111–3119 (2013)
26. Mitra, B., Diaz, F., Craswell, N.: Learning to match using local and distributed representations of text for web search. In: *Proceedings of WWW*, pp. 1291–1299 (2017)
27. Nagel, S.S.: Applying correlation analysis to case prediction. *Texas Law Rev.* **42**, 1006 (1963)
28. Nguyen, T., et al.: Ms marco: a human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268* (2016)
29. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: Squad: 100,000+ questions for machine comprehension of text. In: *Proceedings of EMNLP* (2016)
30. Rocktaschel, T., Grefenstette, E., Hermann, K.M., Kocisky, T., Blunsom, P.: Reasoning about entailment with neural attention. In: *Proceedings of ICLR* (2016)
31. Segal, J.A.: Predicting supreme court cases probabilistically: The search and seizure cases, 1962–1981. *Am. Polit. Sci. Rev.* **78**(4), 891–900 (1984)
32. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *JMLR* **15**(1), 1929–1958 (2014)
33. Sulea, O.M., Zampieri, M., Vela, M., Genabith, J.V.: Exploring the use of text classification in the legal domain. In: *Proceedings of ASAIL workshop* (2017)
34. Tang, D., Qin, B., Liu, T.: Document modeling with gated recurrent neural network for sentiment classification. In: *Proceedings of EMNLP*, pp. 1422–1432 (2015)
35. Ulmer, S.S.: Quantitative analysis of judicial processes: some practical and theoretical applications. *Law Contemp. Probs.* **28**, 164 (1963)
36. Wang, S., Jiang, J.: Learning natural language inference with LSTM. In: *Proceedings of NAACL* (2016)
37. Wang, W., Yang, N., Wei, F., Chang, B., Zhou, M.: Gated self-matching networks for reading comprehension and question answering. *Proc. ACL* **1**, 189–198 (2017)
38. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A.J., Hovy, E.H.: Hierarchical attention networks for document classification. In: *Proceedings of NAACL*, pp. 1480–1489 (2016)
39. Zhong, H., Guo, Z., Tu, C., Xiao, C., Liu, Z., Sun, M.: Legal judgment prediction via topological learning. In: *Proceedings of EMNLP* (2018)
40. Zhuge, P.: *Chinese Law Yearbook*. The Chinese Law Yearbook Press, Shanghai (2016)