Judgment Prediction Based on Case Life Cycle

Luyao Ma
National Engineering Research Center for Software Engineering, Peking University
1701210338@pku.edu.cn

Wei Ye*
National Engineering Research Center for Software Engineering, Peking University
wye@pku.edu.cn

Shikun Zhang
National Engineering Research Center for Software Engineering, Peking University
zhangsk@pku.edu.cn

ABSTRACT

Legal judgment prediction(LJP) is an essential task for legal AI. While prior methods employed the static judge-summarized case narrative as the only input, neglecting critical case life-cycle information could threaten the case logic representation quality and prediction correctness. In this paper, we approach to predict the legal judgment in a reasonably encyclopedic manner by leveraging the genuine input of the case – plaintiff's claims and court debate data with comprehensively understanding the multi-role dialogues of the court debate, and then learnt to discriminate the claims so as to reach the final judgment through multi-task learning. An extensive set of experiments with a large civil trial data set shows that the proposed model can more accurately characterize the interactions among claims, judgments and debate for legal judgment prediction against several alternatives while the neural predictions can also be interpretable and easily observed.

CCS CONCEPTS

• **Computing methodologies → Discourse, dialogue and pragmatics**; **Multi-task learning**; • **Applied computing → Law**.

KEYWORDS

judgment prediction, case life-cycle, multi-task learning

1 INTRODUCTION

Legal judgment prediction has been originally proposed in 1960s (entitled "Using Simple Calculations Predict Judicial Decisions [10]"), unfortunately, prior studies [1, 3, 4, 7, 8, 14, 17, 18] ignored the multi-stage nature of the legal case. In this study, in order to recover the case jigsaw puzzle, we propose an innovative neural model to integrate pre-court claims and court debate.

For a litigation process, a case life-cycle often experiences two critical stages for legal AI system: claim collection stage (e.g., plaintiff provides narrative to judge for the target case) and court debate stage (e.g., plaintiff, defendant, witness, lawyer and judge debate on the court focusing on the claims). A comprehensive case life-cycle representation learning can be nontrivial for legal prediction. To achieve this goal, the biggest challenge lies in the difficulty of accurately representing the multi-role court debate, where different camps may not necessarily share the same vocabulary space, and classical NLP algorithms can hardly consume this variation. For instance, the judge can be more responsible for investigating the facts and reading the court rules while the other litigants answer the questions from the judge. Moreover, with opposite position, plaintiff and defendant's attitudes, sentiments and descriptions to the same topic can be quite different. The last but not the least is the challenge of representing the relations/interactions among the debate, claims and judgment. In civil cases, the judgment can be generalized as the answer to the claims while it is common to have multiple claims in one case and whether they are established or not is not relatively independent.

Motivated by such observations, in this paper, we propose a novel neural automatic judgment prediction model which can predict the judgment result of each claim. Fig. 1 depicts the systematic structure of the proposed model. In a joint learning process, various kinds of information collected from different stages, e.g., court debate, claims, are encapsulated to regularize the judgment prediction (claim classifier) of the civil case.

To the best of our knowledge, this work is the first pioneer investigation of judgment prediction through the life-cycle case data, and our contributions are as follows:

(1) We propose an end-to-end model in a manner of multi-task learning process for a novel legal intelligence problem to address the judgment prediction by exploring the interactions between court debate and plaintiff's claims.

(2) We evaluate the proposed model via $70k$ court records of civil trials along with their judgment documents. The experimental results demonstrate the proposed approach significantly improves the performance of judgment prediction of lawsuit cases against several alternatives.

2 OUR APPROACH

In this section, as Fig. 1 depicts, we present our four-fold automatic judgment prediction framework by leveraging case life-cycle data: (1) the model takes a court debate and its pre-court claims as input, which are encoded by a hierarchical dialogue encoder and a claim encoder respectively. (2) We model the interaction between utterances and claims, as well as the interaction across claims to enhance the claim representations. (4) After multi-hop updates, we feed the final claim representations to a multi-class classifier.

2.1 Input Module

2.1.1 Debate Utterance Encoder. Given an utterance U_i with l words $S_i = \left\{w_{i1}^u, w_{i2}^u, \cdots, w_{il}^u\right\}$ and the role of its speaker $r_i \in R$, we

[*] Corresponding author.

LegalAI'20@SIGIR'20, July 30, 2020, Virtual Event, China
© 2020 Association for Computing Machinery.
ACM ISBN TBA…$15.00
https://doi.org/TBA

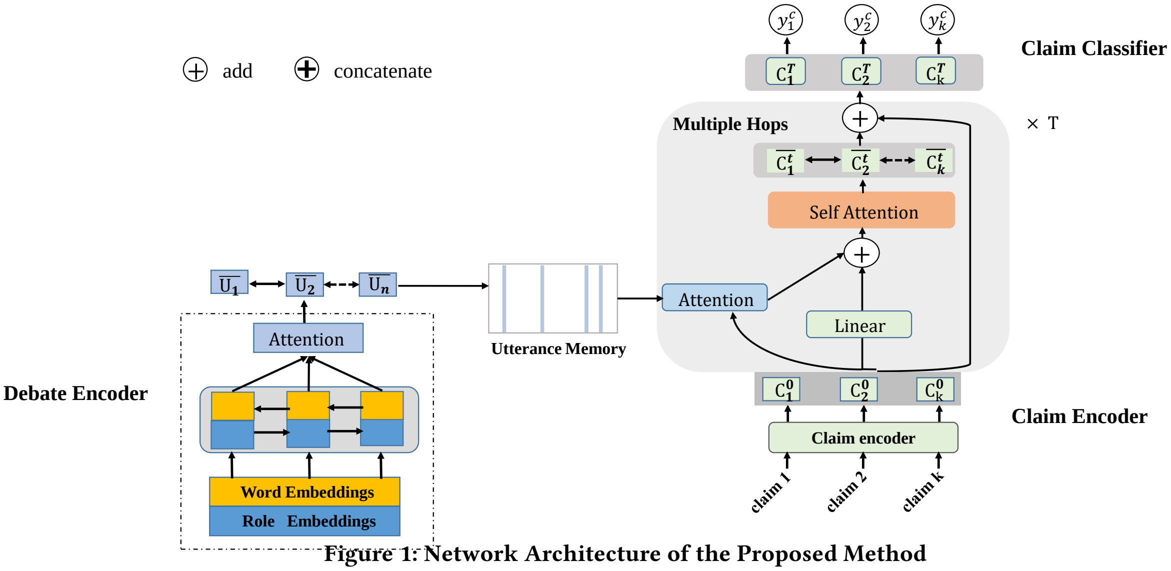

Figure 1: Network Architecture of the Proposed Method

first embed the words to vectors to obtain $\hat{S}_i = \left\{ \mathrm{w}^u_{i1}, \mathrm{w}^u_{i2}, \cdots, \mathrm{w}^u_{il} \right\}$ where $\mathrm{w}^u \in \mathbb{R}^d$ and employ role embedding to encode the role. The role embedding $\mathrm{e}^r_i \in \mathbb{R}^r$ is randomly initialized and jointly learnt during the training process.

To involve the role information into the utterance, we concatenate the role information with each word in the utterance, which is able to project the same word into different dimensional spaces w.r.t.the target role. We hypothesize that the same word may need differentiate when different speakers use it.

$$\mathrm{e}^u_{it} = \mathrm{w}^u_{it} \oplus \mathrm{e}^r_i, \ t \in [1, l] \tag{1}$$

where $\oplus$ denotes a concatenation operation and then the dimention of e^u_{it} is $(d + r)$.

Then we utilize a bidirectional-LSTM to encode the semantics of the utterance while maintaining its syntactic.

$$\begin{aligned} \overrightarrow{\mathrm{h}^u_{it}} &= \overrightarrow{\mathrm{LSTM}^U}(\mathrm{e}^u_{it}), \ t \in [1, l] \\ \overleftarrow{\mathrm{h}^u_{it}} &= \overleftarrow{\mathrm{LSTM}^U}(\mathrm{e}^u_{it}), \ t \in [l, 1] \\ \mathrm{h}^u_{it} &= \overrightarrow{\mathrm{h}^u_{it}} \oplus \overleftarrow{\mathrm{h}^u_{it}} \end{aligned} \tag{2}$$

where h^u_{it} is the concatenation of the forward hidden state of $\overrightarrow{\mathrm{h}^u_{it}}$ and backward hidden state $\overleftarrow{\mathrm{h}^u_{it}}$.

To strengthen the relevance between words in an utterance, we employ the attention mechanism to obtain U_i, which can be interpreted as a local representation of an utterance:

$$\begin{aligned} \mathrm{U}_i &= \sum_{t=1}^{l} \alpha^u_{it} \mathrm{h}^u_{it} \\ \alpha^u_{it} &= \frac{\exp(\mathrm{Q}^u \mathrm{h}^u_{it})}{\sum_{t=1}^{l} \exp(\mathrm{Q}^u \mathrm{h}^u_{it})} \end{aligned} \tag{3}$$

where Q^u are learnable parameters and all parameters in utterance encoder are shared across utterances.

2.1.2 *Debate Dialogue encoder.*

To represent the global context in a dialogue, we use another bidirectional-LSTM to encode the dependency between utterances to obtain a global representation of an utterance, denoted as $\overline{\mathrm{U}_i}$.

$$\begin{aligned} \overrightarrow{\mathrm{h}_i} &= \overrightarrow{\mathrm{LSTM}^D}(\mathrm{U}_i), \ i \in [1, n] \\ \overleftarrow{\mathrm{h}_i} &= \overleftarrow{\mathrm{LSTM}^D}(\mathrm{U}_i), \ i \in [n, 1] \\ \overline{\mathrm{U}_i} &= \overrightarrow{\mathrm{h}_i} \oplus \overleftarrow{\mathrm{h}_i} \end{aligned} \tag{4}$$

where $\overline{\mathrm{U}_i}$ is the i-th utterance's global representation.

2.1.3 *Pre-court Claim Encoder.*

Similar to the input utterances, we encode the claims via bidirectional-LSTM and use attention mechanism to obtain the local representations of the claim. We share the word embedding matrix across the utterance encoder and the claim encoder.

$$\begin{aligned} \overrightarrow{\mathrm{h}^c_{jv}} &= \overrightarrow{\mathrm{LSTM}^C}(\mathrm{w}^c_{jv}), \ v \in [1, q] \\ \overleftarrow{\mathrm{h}^c_{jv}} &= \overleftarrow{\mathrm{LSTM}^C}(\mathrm{w}^c_{jv}), \ v \in [q, 1] \\ \mathrm{h}^c_{jv} &= \overrightarrow{\mathrm{h}^c_{jv}} \oplus \overleftarrow{\mathrm{h}^c_{jv}} \\ \mathrm{C}_j &= \sum_{v=1}^{q} \alpha^c_{jv} \mathrm{h}^c_{jv} \\ \alpha^c_{jv} &= \frac{\exp(\mathrm{Q}^c \mathrm{h}^c_{jv})}{\sum_{v=1}^{q} \exp(\mathrm{Q}^c \mathrm{h}^c_{jv})} \end{aligned} \tag{5}$$

where C_j is the j-th claim's representation and Q^c are learnable parameters and the parameters in claim encoder are shared across claims.

2.2 Interaction Module

2.2.1 Debate-to-Claim. Utterance vectors are stacked and regarded as an utterance memory $\mathrm{m}^u = \left\{\overline{\mathrm{U}_1}, \overline{\mathrm{U}_2}, \cdots, \overline{\mathrm{U}_n}\right\}$. We compute attention weights where each weight indicates the correlation between a claim vector C_j and an utterance memory unit m_i^u.

$$\begin{aligned} \mathrm{O}_j^u &= \sum_{i=1}^{n} \alpha_{ji}^d \overline{\mathrm{U}_i} \\ \alpha_{ji}^d &= \frac{\exp(\mathrm{C}_j \overline{\mathrm{U}_i})}{\sum_{i=1}^{n} \exp(\mathrm{C}_j \overline{\mathrm{U}_i})} \end{aligned} \tag{6}$$

where O_j^u is the output vector of the interaction between utterance memory and a claim.

2.2.2 Fusion. For each claim, we obtain the output O_j^u from utterance memory. We further apply a linear layer with Rectifier Liner Unit (ReLU) to obtain $\hat{\mathrm{C}_j}$. After the addition, we get $\overline{\mathrm{C}_j}$ as the claim representation via memory blocks.

$$\begin{aligned} \hat{\mathrm{C}_j} &= \mathrm{ReLU}(W^l \mathrm{C}_j + b^l) \\ \overline{\mathrm{C}_j} &= \hat{\mathrm{C}_j} + \mathrm{O}_j^u \end{aligned} \tag{7}$$

where W^u, W^f, W^l, b^g and b^l are trainable parameters shared across claims.

2.2.3 Across-Claim. As aforementioned it is common to have multiple claims in one case and whether they are established or not is not relatively independent, it is necessary to model the dependency across claims. Technically, we employ self attention mechanism to capture the relationships across claims.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{8}$$

where $Q \in \mathbb{R}^{d_k \times k}, K \in \mathbb{R}^{d_k \times k}, V \in \mathbb{R}^{d_k \times k}$ are query, key, value which are the same vector.

We take the stack of claim vectors $\overline{\mathrm{C}^c} = \left\{\overline{\mathrm{C}_1}, \overline{\mathrm{C}_2}, \cdots, \overline{\mathrm{C}_k}\right\}$ as input to a self-attention layer with residual connections.

Moreover, we employ multiple (T) hops (denoted as the grey block in Fig. 1) in our model where the output of the previous hop is considered as the input of next hop. Previous works [11, 13] have proved the usage of multiple hops in memory network could yield learn the deep abstraction of text.

2.3 Judgment Prediction

After T hops updates, we obtain the final claim representation C_j^T for j-th claim and feed it to the softmax layer for judgment prediction.

$$y_j^c = \mathrm{softmax}(W^c \mathrm{C}_j^T + b^c) \tag{9}$$

We train our model in an end-to-end manner by minimizing the cross-entropy loss.

$$\mathcal{L}_c = -\frac{1}{k} \sum_{j=1}^{k} \sum_{d=1}^{|Y_c|} g_{jd}^c \log(y_{jd}^c) \tag{10}$$

where g_{jd}^c, y_{jd}^c are the ground truth and the predicted probability of d-th class for j-th claim for each training instance, respectively.

3 EXPERIMENT SETTINGS

3.1 Dataset Construction

In the experiment, we collected 70, 482 cases of Debt Collection category. Each case includes plaintiff's claims, court debate records and judgment document. In total, it contains more than 4.1 million utterances and 133, 209 claims. On average, each case contains 58.17 utterances and 1.89 claims. The ratio of category labels[1] in main task is 1 : 2.6 : 10.9.

3.2 Training Details

The dimensions of word embeddings and role embeddings are set to 300. Word embeddings are trained using the Skip-Gram model[9] on the debate dialogues and role embeddings are randomly initialized. The size of hidden states of bidirectional-LSTM is 256. The neural networks are trained using Adam Optimization[6] with a learning rate set to 0.001, and perform the mini-batch gradient descent with a batch size of 16. The dropout is set to 0.8.

4 RESULT DISCUSSION

4.1 Evaluation Metrics

We use Macro F_1 and Micro F_1 (Mac.F_1 and Mic.F_1 for short) as the main metrics for algorithm evaluation. In a multi-class classification setup, macro-average reflects the robustness of the model if there exists class imbalance. Note that as for all the baselines, we set the debate content concatenated with each claim of the case as input[2] and the judgment result for each claim as output. As for the proposed methods, we are capable of predicting the judgment results of all the claims of a case at once, thus in each sample we conduct k multi-class classification tasks where k is the number of claims in a case.

4.2 Baselines

To extensively validate the effectiveness of the proposed model, several baselines are employed:

Traditional machine learning based method. **TFIDF+SVM** is a robust multi-class classification by means of TFIDF and SVM [12].

Deep learning based methods. **TextCNN** is a convolutional neural networks trained on top of pre-trained word vectors[3] for sentence-level classification tasks [5] where the entire content of the debate is regarded as input.

BiGRU+ATT employs Bi-directional GRU with attention mechanism [15] to capture context semantics and automatically selects important features through attention during training, which is a variant of attention-based RNNs. Similar to the method **TextCNN**, we use entire debate content as input.

HAN stands for Hierarchical Attention Network [16] which is a hierarchical text classification model with two levels of attention mechanisms for aggregating words to utterance and utterances to dialogue.

BERT [2] is a fine-tuning representation model. We take the representation of "[CLS]" as aggregated representation and add a softmax layer on the top of BERT for judgment prediction.

[1] The labels of main task classifier: *reject*, *partially support* and *support*

[2] If a case contains k claims, then such case forms to k samples in which each sample input is the combination of the debate content and one of the k claims.

[3] Skip-gram model[9] is utilized for pretraining word representations.

Method	Reject			Partially Support			Support			Average			
	P	R	F1	P	R	F1	P	R	F1	Mac.P	Mac.R	Mac.F1	Mic.F1
TFIDF+SVM	**77.3**	29.9	43.1	57.6	36.0	44.3	83.0	**94.7**	88.5	72.6	53.6	58.7	79.9
TextCNN	68.3	48.1	56.5	58.6	45.2	51.0	85.7	92.5	88.9	70.9	62.0	65.5	81.1
BiGRU-ATT	73.5	57.1	64.3	66.3	49.5	56.7	86.8	93.7	90.1	75.5	66.7	70.3	83.3
HAN	72.1	60.4	65.7	68.9	46.1	55.2	86.5	94.5	90.4	75.8	67.0	70.4	83.6
BERT	70.7	51.0	59.3	62.6	51.7	56.6	86.8	92.5	89.6	73.4	65.1	68.5	82.5
Ours (static)	73.8	46.4	57.0	66.2	51.0	57.6	86.5	93.9	90.0	75.5	63.8	68.2	83.1
Ours	73.6	59.5	65.8	65.6	**62.7**	64.1	**89.7**	92.0	90.8	76.3	71.4	73.6	84.6

Table 1: Main Results of All Tested Methods for the Main Task. Note that the average scores shown at row Ours are statistically significant different from the corresponding value of all the baseline models (p-value<0.001).

4.3 Overall Performance

To evaluate the performance of the proposed model, we export the results from four perspectives:

Comparison Against Baselines. Table 1 summarizes the performance of all the tested methods over the possible classes of a claim being judged of the main task. We have the following observations from the results: (1) It is not surprising to see that the traditional machine learning based methods didn't perform well in terms of F_1 scores. It indicates the importance of legal case representation learning for better judgment prediction. (2) Among the deep learning based baselines, **HAN** outperforms the other "single-level" models for both macro F_1 and micro F_1 scores which indicates the necessity of using hierarchical context representation to capture the dependency within words, utterance, and dialogue in the court debate scenario. In addition, **BERT** shows lower performance due to its maximum length limitation in long dialogue modeling as well as the ignorance of the role information. (3) **Ours** outperforms all the other tested methods over both macro F_1 and micro F_1 metrics. Moreover, we can observe that **Ours-MTL** are less sensitive to the low frequency categories ("partially support" and "reject") where it performs better than the best baseline by a bigger margin.

Static Case Narrative vs Case Life-Cycle Learning. As aforementioned, a legal case often experiences different stages, and a snippet from court debate may overrule the initial decisions. By comparing static case narrative **Ours (static)** (using case claims as the only input), we can clearly sense the significance of court debate data (**Ours**) for legal decision making. This improvement confirms that properly integrating claims and court debate can indeed enhance the algorithm performance. Intuitively, judges can make wise decisions only if they can comprehensively investigate the information collected from multiple stages and different camps of litigants. Experiment result verifies that projecting static claims to dynamic debate can be important for this task.

5 CONCLUSION

Performing case life-cycle admissibility inspection over court debate dialogues can be practically useful to assist the judges to adjudicate cases. In this work, we introduce a delicately designed life-cycle case representation technique and provide an end-to-end model in a manner of multi-task learning process. The empirical findings validate our hypothesis can improve the performance over state-of-the-art approaches. Additionally, our model is capable of discovering the mutual effect among different features: claims, debates and judgments.

REFERENCES

[1] Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. 2019. Neural Legal Judgment Prediction in English. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 4317–4323.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 4171–4186.

[3] Charles M Haar, John P Sawyer Jr, and Stephen J Cummings. 1977. Computer power and legal reasoning: A case study of judicial decision prediction in zoning amendment cases. *Law & Social Inquiry* 2, 3 (1977), 651–768.

[4] Xin Jiang, Hai Ye, Zhunchen Luo, WenHan Chao, and Wenjia Ma. 2018. Interpretable rationale augmented charge prediction system. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*. 146–151.

[5] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1746–1751.

[6] Diederick P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

[7] Reed C Lawlor. 1963. What computers can do: Analysis and prediction of judicial decisions. *American Bar Association Journal* (1963), 337–344.

[8] Bingfeng Luo, Yansong Feng, Jianbo Xu, Xiang Zhang, and Dongyan Zhao. 2017. Learning to Predict Charges for Criminal Cases with Legal Basis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2727–2736.

[9] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[10] Stuart Nagel. 1960. Using simple calculations to predict judicial decisions. *American Behavioral Scientist* 4, 4 (1960), 24–28.

[11] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*. 2440–2448.

[12] Johan AK Suykens and Joos Vandewalle. 1999. Least squares support vector machine classifiers. *Neural processing letters* 9, 3 (1999), 293–300.

[13] Duyu Tang, Bing Qin, and Ting Liu. 2016. Aspect Level Sentiment Classification with Deep Memory Network. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 214–224.

[14] Frederick Bernays Wiener. 1962. Decision prediction by computers: Nonsense cubed—and worse. *American Bar Association Journal* (1962), 1023–1028.

[15] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*. 2048–2057.

[16] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*. 1480–1489.

[17] Hai Ye, Xin Jiang, Zhunchen Luo, and Wenhan Chao. 2018. Interpretable Charge Predictions for Criminal Cases: Learning to Generate Court Views from Fact Descriptions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 1854–1864.

[18] Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Chaojun Xiao, Zhiyuan Liu, and Maosong Sun. 2018. Legal judgment prediction via topological learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 3540–3549.