Legal Charge Prediction via Bilinear Attention Network

Yuquan Le
Hunan University
Changsha, China
leyuquan@hnu.edu.cn

Yuming Zhao
JD AI Research
Beijing, China
miller21th@gmail.com

Meng Chen
JD AI Research
Beijing, China
chenmeng20@jd.com

Zhe Quan
Hunan University
Changsha, China
quanzhe@hnu.edu.cn

Xiaodong He
JD AI Research
Beijing, China
xiaodong.he@jd.com

Kenli Li
Hunan University
Changsha, China
lkl@hnu.edu.cn

ABSTRACT

The legal charge prediction task aims to judge appropriate charges according to the given fact description in cases. Most existing methods formulate it as a multi-class text classification problem and have achieved tremendous progress. However, the performance on low-frequency charges is still unsatisfactory. Previous studies indicate leveraging the charge label information can facilitate this task, but the approaches to utilizing the label information are not fully explored. In this paper, inspired by the vision-language information fusion techniques in the multi-modal field, we propose a novel model (denoted as LeapBank) by fusing the representations of text and labels to enhance the legal charge prediction task. Specifically, we devise a representation fusion block based on the bilinear attention network to interact the labels and text tokens seamlessly. Extensive experiments are conducted on three real-world datasets to compare our proposed method with state-of-the-art models. Experimental results show that LeapBank obtains up to 8.5% Macro-F1 improvements on the low-frequency charges, demonstrating our model's superiority and competitiveness.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**;
• **Applied computing** → **Law**.

KEYWORDS

Charge Prediction, Bilinear Attention Network, Label Embedding, Legal Artificial Intelligence

ACM Reference Format:
Yuquan Le, Yuming Zhao, Meng Chen, Zhe Quan, Xiaodong He, and Kenli Li. 2022. Legal Charge Prediction via Bilinear Attention Network. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3511808.3557379

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557379

1 INTRODUCTION

Legal Artificial Intelligence (LegalAI) [57] aims to help legal tasks with artificial intelligence technology and has become a trendy research field. Charge prediction is one of the critical problem in LegalAI. Given the fact description of cases, charge prediction tries to predict the judgment results (e.g., fraud, theft, or homicide) of legal cases by analyzing law documents, which has significant value for assisting lawyers on real-world legal judgment. For example, it can reduce redundant work for legal experts, and it can also provide legal consulting services for ordinary people who are unfamiliar with legal terminology. Most existing works usually treat legal charge prediction as a **multi-class text classification** problem. Early models involve feature engineering [11, 19–21, 23, 40], which rely on hand-crafted features such as Bag-of-words, TF-IDF, and n-grams to represent the text. The effectiveness of these models is primarily limited by feature extraction. Recently, neural models have been employed progressively in this task because of their superiority in learning distributed text representations [6, 8, 17, 26, 55] and remarkable progress has been made.

Unlike the common text classification problem, there are several particularities for legal charge prediction. In real-world scenarios, the number of charges is usually large (hundreds of labels). The distribution of charges is exceptionally imbalanced, especially the long-tail charges with limited cases tend to be numerous. Taking a real-world dataset[1] for example, according to previous statistics [17], there is a total of 149 charges. However, the top 10 majority charges (e.g., theft, intentional injury) cover 77.8% cases. In comparison, the top 50 minority charges (e.g., reselling artifacts, tax-escaping) only cover less than 0.5% cases, and most of them even contain less than 10 cases, which brings great challenges for the classifier.

Intuitively, when professional lawyers make the legal judgement, they not only look at the text of fact description, but also need to understand the semantic meanings of each charge label. By combining the information of both sides organically, the understanding becomes more thorough. Previous studies [4, 24, 46, 47] also point out leveraging the linguistic knowledge of charge labels can facilitate this task. The charges in the legal area are usually well-defined, and each charge label can be treated as an accurate and refined description [24]. Then, the informative charge labels can potentially guide the model to focus on the most salient information in the fact

[1]The dataset was published by China Judgments Online, and can be downloaded from: https://thunlp.oss-cn-qingdao.aliyuncs.com/attribute_charge.zip

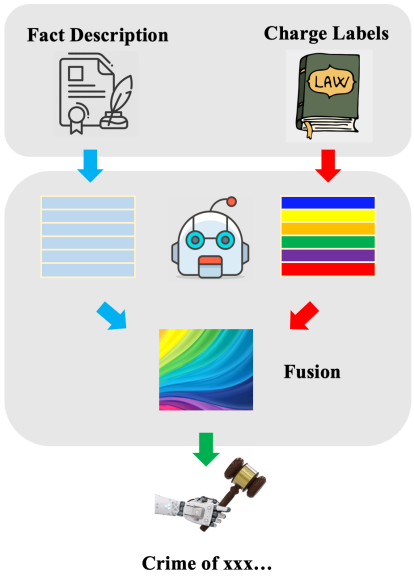


Figure 1: Legal charge prediction by fusing the representations of text and labels.

description [46]. Most existing methods usually use dot-product [4, 47] or cosine similarity [46] to obtain the matching scores between labels and words. However, the rich knowledge of charge labels have not yet been fully explored.

Therefore, we argue that fusing the information from different sources (fact description and charge labels) into a single expressive representation is crucial. As Figure 1 shown, the process can be analogous to vision-language information fusion in the multi-modal field. Imagining charge labels as the image regions/objects in the visual channel and the fact description as the language channel, the model aims to exploit the fine-grained interactions between two groups of input channels. Bilinear pooling-based fusion, also referred to as second-order pooling [43], is a popular method to fuse visual feature vectors with textual feature vectors to create a joint representation space by computing their outer product [53], which facilitates multiplicative interactions between all elements in both vectors. Compared with simple vector combination (e.g., weighted sum, element-wise multiplication, or concatenation), bilinear pooling can generate a more expressive representation by linearizing the matrix from the outer product into a vector. Recently, Bilinear Attention Networks (BAN) [13] have been proposed and shown more intriguing properties on exploiting bilinear interactions between two groups of input channels by combining the advantages of the bilinear pooling and attention mechanism. By decomposing the interaction weight matrix into low-rank modality-specific factors [2, 52], BAN significantly reduces the number of parameters and improves efficiency. After equipped with attention mechanism, a bilinear attention map representing an attention distribution is produced to fuse the vision-language information seamlessly.

Inspired by this, in this paper, we propose a novel model (denoted as **LeapBank**), equipped with a Bilinear Attention Fusion (BAF) block, to catch the fine-grained bilinear interactions and fuse the information between given fact description and charge labels seamlessly. To validate the effectiveness of LeapBank, we experiment on three public real-world charge prediction datasets, and show that it outperforms existing baselines and create new state-of-the-art results. To summarize, the main contributions of this paper are threefold:

- We propose a novel model (LeapBank) for the legal charge prediction task. Our model devises a bilinear attention fusion block to utilize given charge-text information seamlessly.
- We conduct extensive experiments to study the performance of the LeapBank based on three benchmark datasets. The experimental results demonstrate that LeapBank leads to 8.5% points improvement of Macro-F1 on the low-frequency charges.
- We also verify the universality of our model by experimenting on a public multi-label legal document classification task. Besides, more representation fusion methods are also compared to show the superiority of BAF.

2 PRELIMINARIES

To help understand our proposed model, here we review the evolution of bilinear models. As a representative approach of information fusion, bilinear related models provide expressive representations, and have achieved attractive performance in the multi-modal field [53]. Given $\mathbf{x} \in \mathbb{R}^{d_x}$ and $\mathbf{y} \in \mathbb{R}^{d_y}$ are two different feature vectors, bilinear models consider all pairwise interactions among features through outer product [18]. The formula is as follows:

$$f_i = \mathbf{x}^T \mathbf{W}_i \mathbf{y}, \tag{1}$$

where $\mathbf{W}_i \in \mathbb{R}^{d_x \times d_y}$ is a weight matrix for the output f_i. Please note for K output features, the number of parameters of $\mathbf{W}$ is $K \times d_x \times d_y$.

For better regularization, the low-rank bilinear method is introduced to reduce the rank of the weight matrix $\mathbf{W}_i$ [36]. It can be formulated as Equation 2:

$$f_i = \mathbf{x}^T \mathbf{W}_i \mathbf{y} \approx \mathbf{x}^T \mathbf{U}_i \mathbf{V}_i^T \mathbf{y} = \mathbb{1}^T (\mathbf{U}_i^T \mathbf{x} \circ \mathbf{V}_i^T \mathbf{y}), \tag{2}$$

where $\mathbf{W}_i$ is replaced with two smaller matrices $\mathbf{U}_i \in \mathbb{R}^{d_x \times d}$ and $\mathbf{V}_i \in \mathbb{R}^{d_y \times d}$. $\mathbf{x} \in \mathbb{R}^{d_x}$ denotes the textual feature vectors and $\mathbf{y} \in \mathbb{R}^{d_y}$ represents the visual feature vectors. d_x and d_y represent the dimension of corresponding vectors. $\mathbb{1} \in \mathbb{R}^d$ is a vector of ones and $\circ$ denotes Hadamard product (element-wise multiplication).

However, for output feature $\mathbf{f}$, the two third-order tensors ($\mathbf{U}$ and $\mathbf{V}$) are still needed. To reduce the number of parameters, Kim et al. [14] propose low-rank bilinear pooling by introducing a pooling matrix $\mathbf{P}$, which allows $\mathbf{U}$ and $\mathbf{V}$ to be two-order tensors. The formula is as follows:

$$\mathbf{f} = \mathbf{P}^T (\mathbf{U}^T \mathbf{x} \circ \mathbf{V}^T \mathbf{y}). \tag{3}$$

Recently, Bilinear Attention Network (BAN) [13] generalizes the above bilinear models by exploiting bilinear attention maps. BAN combines the advantage of low-rank bilinear pooling and attention mechanism, thus has the ability to utilize given vision-language information to find bilinear attention distributions seamlessly.

Suppose $\mathbf{X} \in \mathbb{R}^{d_x \times N}$ denotes the language channel (e.g. words in question) and $\mathbf{Y} \in \mathbb{R}^{d_y \times M}$ represents the vision channel (e.g. objects in image), where N and M represent the number of two

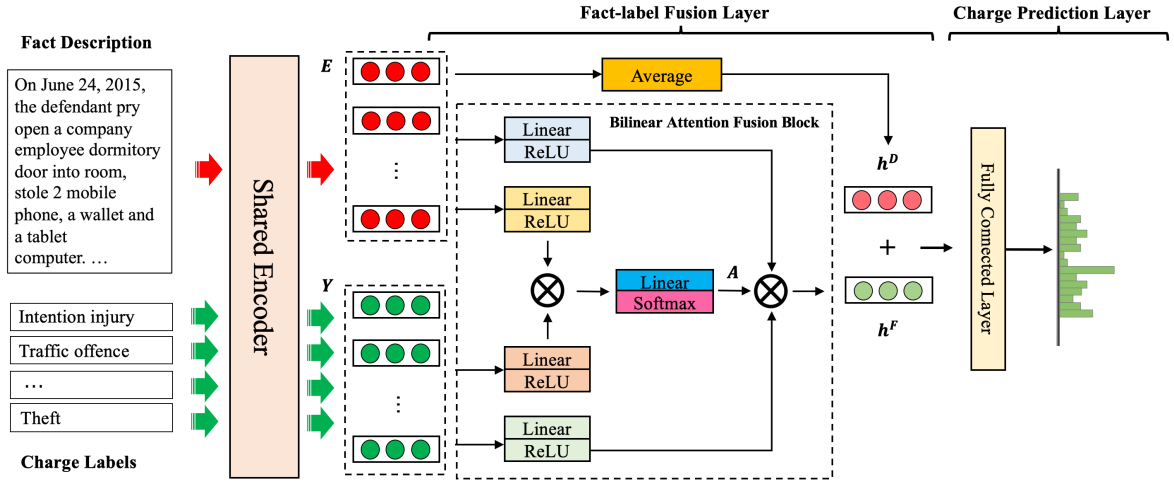


Figure 2: Overview of our model. Fact description and charge labels share the same encoder. E represents the contextual word embeddings of fact description from pre-trained models. Y denotes the embeddings of charges.

input channels. The k-th element of intermediate representation is calculated as follows:

$$\mathbf{f}_k^{'} = (\mathbf{X}^T\mathbf{U}^{'})_k^T\mathbf{A}(\mathbf{Y}^T\mathbf{V}^{'})_k, \tag{4}$$

where $(\mathbf{X}^T\mathbf{U}^{'})_k \in \mathbb{R}^N$ and $(\mathbf{Y}^T\mathbf{V}^{'})_k \in \mathbb{R}^M$. $\mathbf{U}^{'} \in \mathbb{R}^{d_x \times d_k}$ and $\mathbf{V}^{'} \in \mathbb{R}^{d_y \times d_k}$ are weight matrices. $\mathbf{A} \in \mathbb{R}^{N \times M}$ denotes the bilinear attention map, which is calculated as follows:

$$\mathbf{A} = \text{softmax}(((\mathbb{1} \cdot \mathbf{p}^T) \circ \mathbf{X}^T\mathbf{U})\mathbf{V}^T\mathbf{Y}), \tag{5}$$

where $\mathbb{1} \in \mathbb{R}^N$ and $\mathbf{p} \in \mathbb{R}^{d_k}$. The $\mathbf{U} \in \mathbb{R}^{d_x \times d_k}$ and $\mathbf{V} \in \mathbb{R}^{d_y \times d_k}$ are weight matrices.

3 MODEL

In this section, we introduce our proposed model in detail, which is illustrated in Figure 2. It takes the fact description and all charge labels as the input. The output is a predicted charge. The central idea of LeapBank is to exploit fine-grained interactions between all charges and the tokens of fact description.

3.1 Shared Encoder for Fact and Charges

Suppose the fact description of a legal case is a word sequence $\mathbf{x} = \{x_1, x_2, ..., x_N\}$, where N is the length of the sequence. $x_i \in V$ is the i-th word of $\mathbf{x}$, and V denotes the fixed vocabulary. Usually, a specific token [CLS] is inserted as the first token. Formally, the input of fact encoder becomes $\mathbf{x} = \{[CLS], x_1, x_2, ..., x_N\}$. Here, we take the popular pre-trained model BERT [3] as our fact encoder without loss of generality. Note that it's not limited to BERT, and other pre-trained models such as ELMo [35], RoBERTa [22], and XLNet [50] are also feasible. The input embedding is the sum of static word embedding, segment embedding, and position embedding. Then, the input embedding is fed into several Transformer blocks [45] to produce the contextual word embeddings of fact description. The formula is as follows:

$$\mathbf{E} = f_{\text{fact\_enc}}([CLS], x_1, ..., x_N), \tag{6}$$

where $\mathbf{E} \in \mathbb{R}^{(N+1) \times d_h}$ and d_h is the embedding dimension of tokens.

Given all charges $C = [c_1, c_2, ..., c_M]$, the charge encoder maps each charge label into a dense vector based on its description words. The $\mathbf{c}_k = [w_1^k, w_2^k, ..., w_L^k]$ denotes the k-th charge in label set. w_i^k indicates the i-th word of k-th charge. M denotes the number of charges and L represents the maximum length of charge description. The charge representation $\mathbf{Y} \in \mathbb{R}^{M \times d_h}$ can be obtained as follows:

$$\mathbf{y}_k = f_{\text{charge\_enc}}([CLS], w_1^k, w_2^k, ..., w_L^k), \tag{7}$$

where $k \in [1, M]$. We take the same encoder BERT for charge encoding. Each charge label is encoded individually with the shared encoder, and the vector of [CLS] is taken as the initial representation of charge embedding. In order to balance the computational efficiency, we extract the charge embeddings of all charge labels from the charge encoder in advance.

3.2 Fact-label Fusion Layer

Inspired by the bilinear attention network in multi-modal field [13], in this paper, we devise a novel Bilinear Attention Fusion (BAF) block, to capture fine-grained interactions between charge labels and tokens in fact description.

Given the charge embeddings $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_M\}$ and contextual word embeddings $\mathbf{E} = \{\mathbf{e}_0, \mathbf{e}_1, ..., \mathbf{e}_N\}$, we define the bilinear attention fusion block as a function:

$$\mathbf{h}^F = \text{BAF}(\mathbf{E}, \mathbf{Y}; \mathbf{A}), \tag{8}$$

where $\mathbf{h}^F$ is the fused representation, and can be calculated by the following formulas:

$$\mathbf{h}_t^{'} = \sigma(\mathbf{E}\mathbf{U}^{'} + \mathbf{b}_{u'}^T)_t^T\mathbf{A}\sigma(\mathbf{Y}\mathbf{V}^{'} + \mathbf{b}_{v'}^T)_t, \tag{9}$$

$$\mathbf{h}^F = \mathbf{P}^T\mathbf{h}^{'}, \tag{10}$$

where $\mathbf{U}^{'} \in \mathbb{R}^{d_h \times d_k}$ and $\mathbf{V}^{'} \in \mathbb{R}^{d_h \times d_k}$ are weight matrices. $\mathbf{b}_{u'} \in \mathbb{R}^{d_k}$ and $\mathbf{b}_{v'} \in \mathbb{R}^{d_k}$ are bias. $\mathbf{h}_t^{'}$ denotes the t-th interaction features.

The subscript t for the matrices indicates the index of column. So for $\mathbf{h}^{'} \in \mathbb{R}^{d_k}$, there is at most d_k-rank bilinear pooling. The pooling matrix $\mathbf{P} \in \mathbb{R}^{d_k \times d_h}$ is used to get the bilinear attention fusion representation $\mathbf{h}^F \in \mathbb{R}^{d_h}$. The $\mathbf{A} \in \mathbb{R}^{(N+1)\times M}$ is the bilinear attention matrix, and is calculated as follows:

$$\mathbf{A} = g(((\mathbb{1} \cdot \mathbf{p}^T) \circ \sigma(\mathbf{E}\mathbf{U} + \mathbf{b}_u^T))\sigma(\mathbf{V}^T\mathbf{Y}^T + \mathbf{b}_v)), \tag{11}$$

where $\mathbf{U} \in \mathbb{R}^{d_h \times d_k}$ and $\mathbf{V} \in \mathbb{R}^{d_h \times d_k}$ are weight matrices. $\mathbf{b}_u \in \mathbb{R}^{d_k}$ and $\mathbf{b}_v \in \mathbb{R}^{d_k}$ are bias. σ is a nonlinear activation function (e.g. ReLU [31] or Tanh). $\mathbb{1} \in \mathbb{R}^{N+1}$ and $\mathbf{p} \in \mathbb{R}^{d_k}$. Notice that the **softmax** function g is applied element-wisely. Besides, we further improve the vanilla BAN [13] by removing the multi-glimpse and simplifying the eight attention maps to one, thus BAF is more parameter-efficient and performance-effective.

3.3 Charge Prediction Layer

The output of the BAF is the fused representation among fact description and charges. It is combined with the contextual word embeddings to enhance the representation of fact description. The final aggregated representation $\mathbf{h}_o$ is calculated as follows:

$$\mathbf{h}_o = \mathbf{h}^D \oplus \mathbf{h}^F, \tag{12}$$

where $\oplus$ denotes the element-wise addition operation, and the $\mathbf{h}^D$ is the average of contextual word embeddings $\mathbf{E}$. Then $\mathbf{h}_o$ is fed into a fully connected layer with **softmax** function to predict the probability of charges. The objective function is defined as follows:

$$L_{\text{charge}} = \sum_{i \in D} \text{CE}(p_i, \hat{p_i}), \tag{13}$$

where $\hat{p_i}$ is the predicted label, p_i is the ground-truth label, D is the number of training samples, and $\text{CE}(\cdot, \cdot)$ is the cross entropy function.

4 EXPERIMENTS

In this part, we first describe the experimental setup. Then, we compare our proposed method with more than 10 strong baseline methods. We also conduct an ablation study to investigate the effectiveness of BAF. Besides, we investigate if LeapBank can improve the performance of low-frequency charges. Finally, we visualize the learnt label embeddings to show the advantage of LeapBank.

4.1 Experimental Setup

Datasets. Following prior works [6, 17], we employ three public datasets[1] to validate our proposed model. The datasets are collected from the real world, published by the Chinese government from China Judgments Online. A legal document is usually well-structured and includes several parts such as fact description, court view, and penalty result [8]. The datasets select the fact description of each legal case as input text. The output label is the charge, which is extracted from the penalty results by the regular expressions. The detailed statistics are presented in Table 2. The three datasets are Criminal-S, Criminal-M, and Criminal-L, which include different amounts of cases but the same amount of charges.

Evaluation Metrics. Following previous works [8, 17, 26], we use four metrics commonly used in classification task to evaluate the performance, including Accuracy (**Acc**), Macro Precision (**MP**), Macro Recall (**MR**), and Macro-F1 score (**F1**). For multi-class classification, the MP, MR, and Macro-F1 are calculated as the macro average of the above metrics correspondingly by taking all classes equally important. Therefore, the MP, MR, and Macro-F1 are more fair to reflect the model's performance on the imbalanced dataset, which is exactly the situation in our experiments. Note that the **Acc** can only reflect the overall performance, which might be dominated by the high-frequency charges.

Baselines. We compared LeapBank against a variety of methods, which can be summarized in following three groups.

- **General purpose text classifiers**, including feature engineering method TFIDF-SVM, which takes term-frequency inverse document frequency (TF-IDF) [39] as features and support vector machine (SVM) [41] as classifier; mainstream neural structures such as CNN [15], LSTM [7], and CNN-Capsule [38]; pre-trained model BERT [3], fine-tuned with one additional fully connected layer, where input features are generated by averaging all token representations.
- **Label embedding based classifiers**, including label-word joint embedding model LEAM [46], which uses cosine similarity to get matching scores between words and labels and uses CNN on the matching matrix to get the label-aware attention to weight the text representation; EXAM [4], which uses dot-product to get matching scores between words and labels and aggregates into predictions; LSAN [47], which uses dot-product to get matching scores and directly weights the text representation by matching scores.
- **Models tailored for legal charge prediction**, including: Fact-Law Attention Model [26], which jointly models the charge prediction task and the relevant article extraction task; Attribute-attentive Model [8], which introduces several discriminative attributes of charges to improve the few-shot charges prediction; HLCP [24], which leverages text information of charges with the attention-based sequence to sequence model; SECaps [6], which designs the seq-caps layer and residual attention unit for charge prediction; and SAttCaps [17], which proposes a self-attentive capsule network to tackle this task. To our knowledge, SAttCaps [17] is the state-of-the-art model of this task.

Implementation Details. We use the official Chinese BERT-base model as the fact encoder. The fact descriptions are processed into character sequence, and the maximum input sequence length is set to 512 tokens. The batch-size is set to 16, while the learning rate is set to 2×10^{-5}. Adam optimizer is adopted in our experiments. The best fine-tuning epoch was decided with $\{3, 4, 5\}$ to balance the model performance and the time consumption. In addition, the experimental settings of baselines are as follows. The hyper-parameter settings of the baseline models are consistent with the original papers [6, 8] and the results are collected from [17] except EXAM, LEAM, LSAN and BERT. For EXAM[2], LEAM[3] and LSAN[4], we keep the same settings as the officially released code except maximum sequence length is set to 512. As to BERT, we take the average of contextual word embeddings as final document representation. Note that we also tried to use the representation

[2]https://github.com/NonvolatileMemory/AAAI_2019_EXAM
[3]https://github.com/guoyinwang/LEAM
[4]https://github.com/EMNLP2019LSAN/LSAN

Table 1: Charge prediction results of different models on three datasets. Improvements are statistically significant with $p < 0.05$.

Datasets	Criminal-S				Criminal-M				Criminal-L			
Metrics	Acc	MP	MR	F1	Acc	MP	MR	F1	Acc	MP	MR	F1
TFIDF-SVM	85.8	49.7	41.9	43.5	89.6	58.8	50.1	52.1	91.8	67.5	54.1	57.5
CNN	91.9	50.5	44.9	46.1	93.5	57.6	48.1	50.5	93.9	66.0	50.3	54.7
LSTM	93.5	59.4	58.6	57.3	94.7	65.8	63.0	62.6	95.5	69.8	67.0	66.8
CNN-Capsule	93.3	61.8	61.0	59.8	94.3	69.7	68.0	67.8	95.2	77.1	72.6	73.3
BERT	95.3	70.6	70.8	70.0	96.4	79.2	75.6	76.5	96.8	83.1	79.1	80.1
EXAM	92.4	58.9	56.6	56.3	94.2	64.1	60.0	60.9	94.5	69.5	64.2	64.6
LEAM	92.9	60.0	59.0	58.7	93.2	64.3	57.4	58.9	94.8	74.5	66.8	68.8
LSAN	93.7	58.3	56.1	56.1	94.9	68.0	63.0	64.2	95.7	79.0	71.7	73.9
Fact-Law Attention Model	92.8	57.0	53.9	53.4	94.7	66.7	60.4	61.8	95.7	73.3	67.1	68.6
Attribute-attentive Model	93.4	66.7	69.2	64.9	94.4	68.3	69.2	67.1	95.8	75.8	73.7	73.1
HLCP	-	-	-	-	-	-	-	-	95.9	78.8	73.5	74.7
SECaps	94.8	71.3	70.3	69.4	95.4	71.3	70.2	69.6	96.0	81.9	79.7	79.5
SAttCaps	95.1	74.2	72.4	72.2	96.0	78.2	76.6	76.4	96.4	85.2	81.9	82.5
LeapBank	**95.8**	**77.0**	**77.1**	**76.4**	**96.8**	**81.3**	**79.4**	**79.8**	**97.1**	**85.9**	**83.0**	**83.6**
w/o attention	95.5	74.5	74.8	73.7	96.6	80.5	78.2	78.5	96.9	83.8	80.7	81.2
w/o fact representation	95.3	73.2	72.7	72.0	96.3	79.3	76.7	77.7	96.8	84.2	79.7	80.5

Table 2: The statistics of datasets.

Datasets	Train	Dev	Test	Charges
Criminal-S	61,589	7,755	7,702	149
Criminal-M	153,521	19,250	19,189	149
Criminal-L	306,900	38,429	38,368	149

of [CLS] token directly, but the performance is less effective. We run multiple trials for each experiment, and the average results are reported to avoid bias introduced by randomness.

4.2 Main Results

Table 1 shows the experimental results, which include three parts of content. The first part shows the performance of general-purpose text classification models. BERT beats all popular deep learning neural structures, which demonstrates the effectiveness of pre-trained model. The second part illustrates the performance of state-of-the-art label embedding models including EXAM, LEAM and LSAN. LeapBank outperforms all three label embedding based classifiers significantly, which indicates that our model can utilize the label information more effectively. The third part shows the performances of legal charge prediction models published in previous works[5], which are collected from [17]. It's interesting that although BERT beats SAttCaps on accuracy, it loses Macro-F1 on both Criminal-S and Criminal-L datasets. It indicates that for the numerous long-tail charges, BERT also has limitations. Meanwhile, LeapBank outperforms BERT and SAttCaps on all metrics. Compared to BERT, LeapBank obtains accuracy improvement of 0.5% on Criminal-S, 0.4% on Criminal-M and 0.3% on Criminal-L, which means **10.6%**, **11.1%**, **9.4%** relative error rate reduction correspondingly[6]. Compared to SAttCaps, LeapBank achieves absolute Macro-F1 improvement of **4.2%**, **3.4%** and **1.1%** on three datasets respectively. The results demonstrate LeapBank not only obtains the best overall accuracy, but also performs better on most of the classes.

4.3 Ablation Study

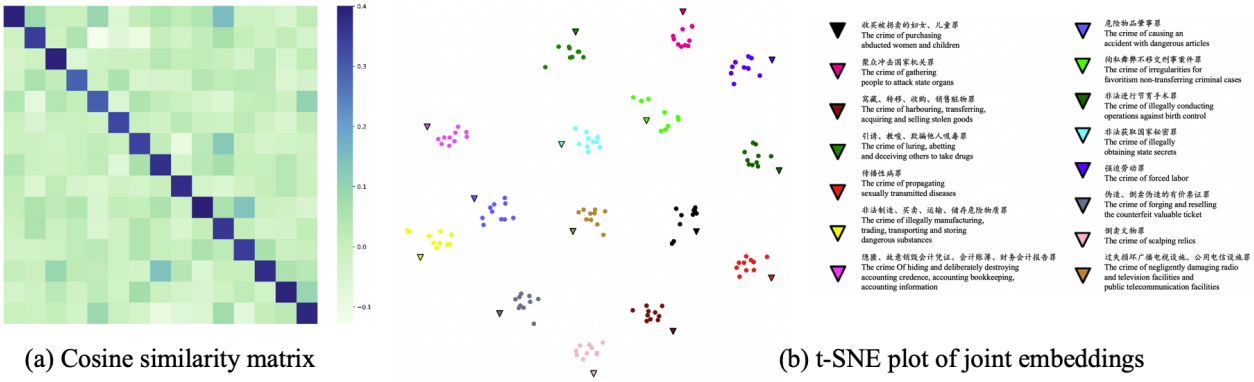
Table 1 also shows the results of ablation study: (1) To figure out the contribution of the bilinear attention map, we remove the bilinear attention map A from Equation 9. After degenerating into the low-rank bilinear form without attention, the performance also degrades on all three datasets. (2) In Equation 12, we aggregate the fact representation $\mathbf{h}^D$ with the bilinear fused representation $\mathbf{h}^F$ to get the final representation. Here, we remove $\mathbf{h}^D$, and use $\mathbf{h}^F$ only. Notice that the performance drops on all datasets, which demonstrates the necessity of the fact representation. Conversely, if we only use the fact representation $\mathbf{h}^D$, the model degenerates into BERT and the performance is also worse than LeapBank. Thus, it indicates that the fact representation $\mathbf{h}^D$ and the bilinear fused representation $\mathbf{h}^F$ are complementary to each other.

4.4 Performance on Low-frequency Charges

To verify the advance of our model on dealing with long-tail charges, we show the performance on charges with different frequencies. We divide the Criminal-S into three parts according to the frequency of charges. Following previous work [8], the charges with ≤ 10 cases are low-frequency (Low). The charges with > 100 cases are high-frequency (High). The others belong to medium-frequency (Medium). Here, we use Macro-F1 metric to evaluate the model

[5]The original paper for HLCP [24] only illustrated the results on Criminal-L, and their source code was not released.

[6]For example, the relative error rate reduction for LeapBank vs. BERT on Criminal-S is calculated as (4.7 - 4.2)/4.7 = 10.6%.



(a) Cosine similarity matrix (b) t-SNE plot of joint embeddings

Figure 3: Correlation between the learned text representation and charge embedding. (a) Cosine similarity matrix between averaged text representation per charge and charge embedding, and (b) t-SNE plot of joint embedding of text and charges.

performance, which can better reflect the performance of the model on the imbalanced scenario.

The experimental results on Criminal-S are shown in Table 3. It shows that LeapBank outperforms all baselines on all frequencies. Especially on low-frequency charges, LeapBank model outperforms SAttCaps by **8.5%** absolute improvement in terms of Macro-F1. We conjecture that there are three possible reasons that leveraging label information can alleviate the data imbalance issue. First, labels with similar semantic meanings may transfer knowledge from each other, so knowledge may be shared across different labels. Second, for the classes with very limited training samples, fusing the auxiliary information from labels can enhance the text representation learning. Third, labels can help the model focus on the most salient information in the lengthy fact description, which can generate more discriminative text representations.

Table 3: Macro-F1 Performance of different models on Criminal-S with different frequencies.

Charge Type	Low	Medium	High
Charge Number	49	51	49
Attribute-attentive Model	49.7	60.0	85.2
SECaps	53.8	65.5	89.0
SAttCaps	59.5	67.8	89.4
LeapBank	**68.0**	**70.7**	**90.7**

4.5 Visualization

To further interpret the experimental results, we visualize the relationship between charge embeddings and document representations in Figure 3. Considering the number of charges is large (total 149 charges), here we focus on visualizing the top 15 minority charges in Criminal-S.

In Figure 3(a), we visualize the cosine similarity matrix between the learnt document representations and charge embeddings. The rows are the averaged per-charge document representations, and the columns are charge embeddings. We can observe that the on-diagonal elements' color is much heavier than the color of the off-diagonal elements. It indicates the learnt charge embeddings can effectively represent the semantic meanings of corresponding charges. In Figure 3(b), we use t-SNE [44] to visualize both the high dimensional representations of the documents and labels on a 2D map. Different charges are depicted by different colors. The dots are the document representations, and the large triangles are the charge embeddings. It's observed that each triangle is nearby the corresponding document representations, while far from other classes. It demonstrates our proposed model can learn very representative embeddings even for the low-frequency charges.

5 DISCUSSION

In this section, we conduct further discussions including: (1) The important hyper-parameter rank k is studied. (2) The other popular fusion approaches are explored. (3) We experiment on a multi-label English legal document classification dataset. (4) We visualize the training process to show BAF can facilitate faster convergence and better accuracy. (5) We replace the shared encoder BERT with other mainstream neural networks to demonstrate BAF can be easily plugged into other neural architectures. (6) We analyze two good cases that LeapBank predicts correctly while BERT makes mistakes. (7) We summarize the limitations of our proposed model and discuss the possible reasons.

5.1 Hyper-parameter Tuning

The rank value k of bilinear pooling is a key hyper-parameter for BAF which influences both the model's performance and efficiency. Here, we experiment on the Criminal-S dataset and tune the hyper-parameter of rank k. Note that the rank value in attention and fusion can be different, but for simplicity, we keep the two rank values consistent to narrow the search space. The result is shown in Table 4. It's observed that larger rank value brings better performance but also contains more model parameters [7]. To balance the performance

[7]Please note that we only report the model parameters of BAF across different ranks k here.

Table 4: Performance of different ranks on Criminal-S

rank	Acc	MP	MR	F1	nParams
96	95.6	72.2	72.2	71.2	0.37M
192	95.6	74.5	74.5	73.8	0.74M
384	95.7	76.6	76.0	75.3	1.48M
768	95.8	77.0	77.1	76.4	2.36M
1536	95.9	76.3	76.2	75.2	5.91M
3072	96.2	78.2	77.4	77.2	11.81M

and the model complexity, we choose $rank = 768$ in our final model.

5.2 Comparison with other Fusion Methods

Table 5: Performance of different fusion methods on Criminal-S

method	Acc	MP	MR	F1
Co-attention	95.5	73.5	73.9	72.9
$\mathbf{BAF}_{concat}$	95.7	76.7	76.4	75.7
$\mathbf{BAF}_{sum}$	95.6	75.1	74.4	73.6
BAF	**95.8**	**77.0**	**77.1**	**76.4**

We conduct further experiments on Criminal-S to replace the key component BAF with other fusion methods. (1) We compare BAF with co-attention [52], which is also a widely applied fusion approach based on attention mechanism in multi-modal field. We apply attention mechanism on labels and document respectively, then fuse the attentive document representation and label embeddings by concatenation. (2) We develop two variant methods $\mathbf{BAF}_{concat}$ and $\mathbf{BAF}_{add}$, which replace the bilinear pooling fusion in BAF with concatenation ($\mathbf{G}_{tij} = [(\mathbf{E}_i\mathbf{U}')_t; (\mathbf{Y}_j\mathbf{V}')_t]$) and summation ($\mathbf{G}_{tij} = (\mathbf{E}_i\mathbf{U}')_t + (\mathbf{Y}_j\mathbf{V}')_t$) respectively. Then the fused representation can be calculated by $\mathbf{h}'_t = \text{sum}(\mathbf{G}_t \circ \mathbf{A})$. For fair comparison, we set the rank to 768. Table 5 illustrates that: (a) BAF and its variants are much better than co-attention, which indicates the necessity of bilinear attention. (b) BAF outperforms $\mathbf{BAF}_{concat}$ and $\mathbf{BAF}_{add}$ by 0.7% and 2.8% points on Macro-F1 respectively, showing the superiority of bilinear pooling fusion.

5.3 Experiment on Multi-label Legal Dataset

To explore the capability of LeapBank on other legal task, we apply our model on the public EUR-Lex dataset [28], which contains 11,585 English legal documents of the European Union and 3,865 testing documents.The task is to predict the legal document into the EUROVOC concept hierarchy with almost 4000 classes. For simplicity, we select the top 500 classes for quick experiment and take BERT-base as our baseline. To adapt the multi-label classification setting, we change the *softmax* to *sigmoid* in the prediction layer. We set the sequence length to 320, learning rate to 5e-5, and batch size to 8 during training, and record the performance every 10,000 train steps until the convergence. Figure 4 shows that LeapBank outperforms BERT significantly on both the accuracy and Macro-F1 during the whole training process and also converges much faster, which demonstrates the generalization ability and advantage of LeapBank on multi-label classification task.

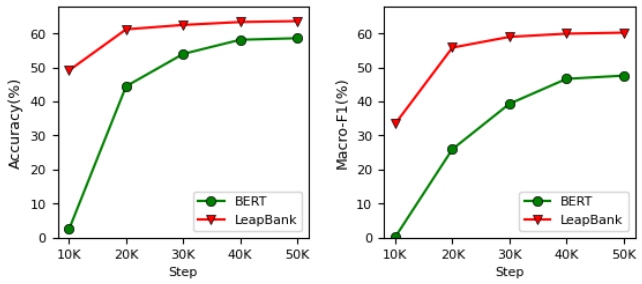
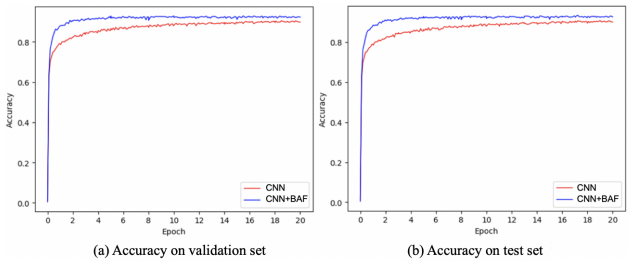


Figure 4: The performance of BERT vs LeapBank on EUR-Lex during the training process.

5.4 Visualization of Model Training



(a) Accuracy on validation set (b) Accuracy on test set

Figure 5: The accuracy changes with the increase of training epochs. (a) Accuracy on validation set, and (b) accuracy on test set.

In Section 5.3, we demonstrate LeapBank can converge much faster and obtain better accuracy than BERT on the multi-label legal dataset EUR-Lex. In this section, we supplement the visualization of model training on the Criminal-S dataset. Without loss of generality, we compare CNN with CNN+BAF. The results are reported in Figure 5. It's observed that: (1) Both the CNN and CNN+BAF can converge within 20 epochs; (2) CNN+BAF converges faster than CNN, and can achieve better accuracy on both the validation set and test set, which indicates that BAF has better generalization ability.

5.5 Universality of BAF

As mentioned in Section 3.1, we utilize the popular pre-trained model BERT [3] as the shared encoder for both fact description and charge labels. Here, we conduct further experiments to show the universality of BAF. To figure out if BAF also works for other neural structures, we plug BAF into more deep learning models (e.g., CNN, GRU, LSTM) and implement corresponding BAF-enhanced models (e.g., CNN+BAF, GRU+BAF, LSTM+BAF). For fair comparison, we keep the model structure and hyper-parameters consistent with

Table 6: Performance of different BAF-enhanced models on Criminal-S.

Model	Acc	MP	MR	F1
CNN	91.9	50.5	44.9	46.1
CNN+BAF	**93.1(↑ 1.2%)**	**56.9(↑ 6.4%)**	**55.0(↑ 10.1%)**	**54.9(↑ 8.8%)**
GRU	93.7	54.4	52.7	52.6
GRU+BAF	**94.5(↑ 0.8%)**	**59.3 (↑ 4.9%)**	**58.0(↑ 5.3%)**	**57.7(↑ 5.1%)**
LSTM	93.5	59.4	58.6	57.3
LSTM+BAF	**94.2(↑ 0.7%)**	**63.2(↑ 3.8%)**	**61.5(↑ 2.9%)**	**61.2(↑ 3.9%)**

our baselines. For the sake of simplicity, we perform experiments on Criminal-S dataset. Table 6 shows that all BAF-enhanced models outperform the corresponding original models significantly on all metrics, which demonstrate that BAF can be an effective module to enhance various neural structures.

5.6 Case Study

In this section, we conduct case study to demonstrate how charge labels help the model make the correct prediction. Figure 6 shows two real cases from the Criminal-S dataset. For both two cases, LeapBank predicted the charge correctly while BERT made mistakes. For the ground-truth charges, both two have only 8 training samples, which means it's quite challenging for the model to learn a discriminative representation from the very limited training data.

For Case (a), the two charges of "Crime of illegally performing birth control operations" and "Illegal practice of medicine" indeed have similar semantic meanings. From the words and phrases marked by the red color, the model is prone to make mistakes because those words indicate the "illegal practice of using drugs". However, because "birth control" occurs in the ground-truth charge label, we can see that the words in the green color (e.g. "fetus" and "birth") guide the model to catch the semantic relationship between fact description and charge labels, which finally facilitates the model's prediction by fusing the information from both sides. Similarly, for Case (b), it can be seen that the words of "forced" and "working" have very similar meaning with the ground-truth charge label ("forced labor"), and finally our proposed model can predict the charge correctly. Both two cases also reveal that leveraging the rich knowledge of charge labels can enhance the representation of fact description and alleviate the data imbalance issue to some extent for this task.

5.7 Error Analysis

However, we also want to analyze the erroneous cases and discuss the limitations of LeapBank. (1) For some cases with lengthy fact description ($\geq$ 1000 characters), our model failed to make correct predictions because the maximum sentence length is set to 512 characters for BERT. (2) For charges with confusion (e.g. "illegally felling trees" and "illegal denudation"), the model failed to distinguish in their nature, especially when both of them are long-tail charges and the charge labels also have similar semantic meanings. (3) For charges with very rare cases (e.g. "destroying computer information system"), there are only 8 cases in Criminal-S, It's still very challenging although LeapBank has alleviated this issue to some extent. We will keep improving above aspects in the future.

6 ETHICAL CONCERNS

The judgment prediction is an emerging technology at its exploratory stage. We should be aware of the risks and prevent any inappropriate use of the technology. There are several ethical concerns worth discussions: 1) The proposed algorithm is designed to predict legal charges for assisting the trial judges for decision making, but it should never replace human judges. Human knowledge/judgment should be the final safeguard to protect social justice and individual fairness. 2) The learnt system may make mistakes because of some subtle details. For example, the numerical values and some infrequent named entities are often hard to learn by neural models, which may cause wrong judgment prediction. The judges need to check the results from algorithm. 3) The model is trained with historical data, which may face potential demographic bias/unfairness challenges, such as gender, age and race bias. Also, with the development of our society, new forms of crimes will appear. The algorithm adoption should be empowered with de-biased legal content pretraining and updated timely.

7 RELATED WORK

7.1 Charge Prediction

Legal charge prediction has drawn attention from researchers in the legal field for decades [12, 16, 27, 30]. Existing works treat charge prediction as a multi-class text classification problem. Both traditional machine learning methods based on feature engineering [11, 19–21, 23, 40] and the recent deep learning based neural models [6, 9, 10, 17, 24–26, 48, 49, 55, 56] are applied in this task and achieve remarkable performance.

Two lines of work are closely relevant to us. One line of works focus on the data imbalance issue and long-tail charges. For example, Hu et al. [8] proposed an attribute-attentive charge prediction model. He et al. [6] presented a sequence enhanced capsule (SECaps) model and devised an attention residual unit to capture crucial textual information. Le et al. [17] designed a self-attentive capsule network, which employs a self-attentive dynamic routing for charge prediction. Different from them, we concentrate on utilizing the linguistic knowledge of charge labels to alleviate this issue. The other line of works propose to utilize the charge label related information to enhance the representation of documents. Liu et al.

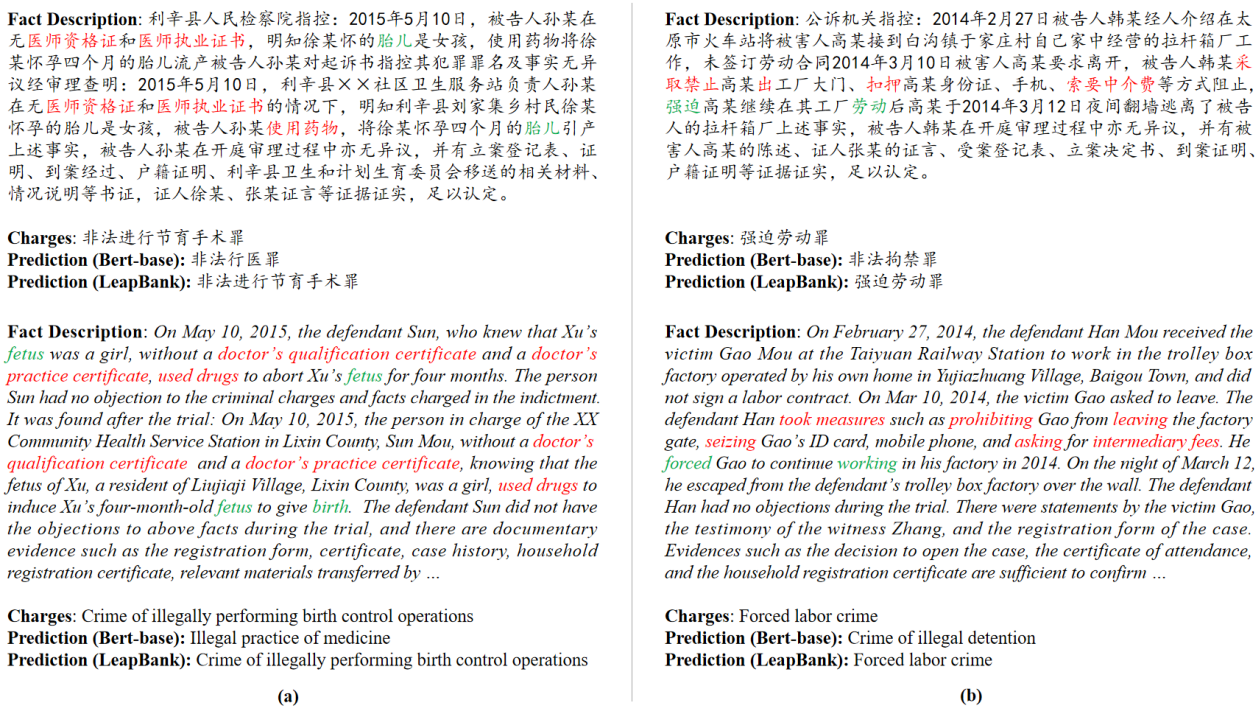


Figure 6: Case study. Words in the green color help model predict correctly. We translate the Chinese to English for better understanding.

[24] incorporated text information of charges with the attention-based sequence to sequence model. Kang et al. [10] explored to use the charge definitions and designed an integrated sentence-level and word-level interaction based on episodic memory attention mechanism. Compared with them: (1) Our model does not need auxiliary expert knowledge or resources (e.g., charge definitions in law or hierarchical dependencies of charges), and thus is easier to extend to other tasks. (2) We analogize the charge labels and fact description to the vision and language channels in the multi-modal field, and formulate their interactions as a multi-modal information fusion problem.

7.2 Label Embedding

Label embedding is to learn the embeddings of the labels in classification tasks and has been proven to be effective in computer vision [1, 5, 37] and natural language processing [29, 32, 42, 46, 51, 54]. One line of related works are Nam et al. [32] and Pappas and Henderson [34]. Nam et al. [32] learned input-label representations by introducing bilinear model and using hinge loss for classification. Pappas and Henderson [34] focused on giving nonlinearity to bilinear function to learn input-label representations with label-set-size independent parametrization. Compared to them, our work also jointly learns the label embeddings. However, they focused on generalizing to unseen labels in classification, while we design a bilinear attention fusion block to fuse the given text-label information seamlessly. Another line of related works are LEAM [46], LSAN [47] and EXAM [4]. The main difference lies in the approach of utilizing label embeddings. LEAM catches the interaction matrix between words and labels based on cosine similarity and applies a convolution layer to measure the attention score for each word. LSAN and EXAM get the word-label similarity matrix by dot product. LSAN directly uses this matrix as weight to get a label-specific document representation, but EXAM puts a MLP on it to get the final prediction. Differently, LeapBank considers all pairwise interactions among tokens in text and labels through outer product (not inner product), thus it can capture fine-grained interactions between text and labels and generate more expressive representation.

8 CONCLUSIONS

This paper proposes a novel model LeapBank to enhance the legal charge prediction task by fusing the information of fact description and charge labels. A novel bilinear attention fusion block is devised to catch the fine-grained interactions between text and labels, which are analogous to vision-language information fusion in multi-modal field. Experimental results on three real-world datasets show that our proposed model outperforms the state-of-the-art baselines significantly. We also verify the universality of LeapBank by experimenting on the multi-label legal document classification task. In the future, we will explore more multi-modal information fusion methods, i.e. X-linear [33], to facilitate this task.

9 ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (No.61907034).

REFERENCES

[1] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. 2016. Label-Embedding for Image Classification. *TPAMI* 38 (2016), 1425–1438.

[2] Hedi Ben-younes, Rémi Cadène, Matthieu Cord, and Nicolas Thome. 2017. MUTAN: Multimodal Tucker Fusion for Visual Question Answering. In *Proc. of ICCV*. 2631–2639.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proc. of NAACL-HLT*. 4171–4186.

[4] Cunxiao Du, Zhaozheng Chen, Fuli Feng, Lei Zhu, Tian Gan, and Liqiang Nie. 2019. Explicit Interaction Model towards Text Classification. In *Proc. of AAAI*. 6359–6366.

[5] Andrea Frome, Gregory S. Corrado, Jonathon Shlens, Samy Bengio, Jeffrey Dean, Marc'Aurelio Ranzato, and Tomás Mikolov. 2013. DeViSE: A Deep Visual-Semantic Embedding Model. In *Proc. of NeuIPS*. 2121–2129.

[6] Congqing He, Li Peng, Yuquan Le, Jiawei He, and Xiangyu Zhu. 2019. SECaps: a sequence enhanced capsule model for charge prediction. In *Proc. of ICANN*. Springer, 227–239.

[7] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[8] Zikun Hu, Xiang Li, Cunchao Tu, Zhiyuan Liu, and Maosong Sun. 2018. Few-Shot Charge Prediction with Discriminative Legal Attributes. In *Proc. of COLING*. 487–498.

[9] Xin Jiang, Hai Ye, Zhunchen Luo, WenHan Chao, and Wenjia Ma. 2018. Interpretable Rationale Augmented Charge Prediction System. In *Proc. of COLING*. 146–151.

[10] Liangyi Kang, Jie Liu, Lingqiao Liu, Qinfeng Shi, and Dan Ye. 2019. Creating auxiliary representations from charge definitions for criminal charge prediction. *ArXiv preprint* abs/1911.05202 (2019).

[11] Daniel Martin Katz, Michael J Bommarito II, and Josh Blackman. 2017. A general approach for predicting the behavior of the Supreme Court of the United States. *PloS one* 12, 4 (2017), e0174698.

[12] R Keown. 1980. Mathematical models for legal prediction. *Computer/LJ* 2 (1980), 829.

[13] Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. 2018. Bilinear Attention Networks. In *Proc. of NeurIPS*. 1571–1581.

[14] Jin-Hwa Kim, Kyoung Woon On, Woosang Lim, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. 2017. Hadamard Product for Low-rank Bilinear Pooling. In *Proc. of ICLR*.

[15] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proc. of EMNLP*. 1746–1751.

[16] Fred Kort. 1957. Predicting Supreme Court Decisions Mathematically: A Quantitative Analysis of the "Right to Counsel" Cases. *American Political Science Review* 51, 1 (1957), 1–12.

[17] Yuquan Le, Congqing He, Meng Chen, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2020. Learning to Predict Charges for Legal Judgment via Self-Attentive Capsule Network. In *Proc. of ECAI*.

[18] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. 2015. Bilinear CNN Models for Fine-Grained Visual Recognition. In *Proc. of ICCV*. 1449–1457.

[19] Wan-Chen Lin, Tsung-Ting Kuo, Tung-Jia Chang, Chueh-An Yen, Chao-Ju Chen, and Shou-de Lin. 2012. Exploiting Machine Learning Models for Chinese Legal Documents Labeling, Case Classification, and Sentencing Prediction. In *International Journal of Computational Linguistics & Chinese Language Processing*, Vol. 17. 49–68.

[20] Chao-Lin Liu, Cheng-Tsung Chang, and Jim-How Ho. 2004. Case instance generation and refinement for case-based criminal summary judgments in Chinese. *JISE* (2004), 783–800.

[21] Chao-Lin Liu and Chwen-Dar Hsieh. 2006. Exploring phrase-based classification of judicial documents for criminal charges in chinese. In *Proc. of ISMIS*. Springer, 681–690.

[22] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv preprint* abs/1907.11692 (2019).

[23] Yi-Hung Liu, Yen-Liang Chen, and Wu-Liang Ho. 2015. Predicting associated statutes for legal problems. *Information Processing & Management* 51, 1 (2015), 194–211.

[24] Zhiyuan Liu, Cunchao Tu, and Maosong Sun. 2019. Legal cause prediction with inner descriptions and outer hierarchies. In *Proc. of CCL*. Springer, 573–586.

[25] Shangbang Long, Cunchao Tu, Zhiyuan Liu, and Maosong Sun. 2019. Automatic judgment prediction via legal reading comprehension. In *China National Conference on Chinese Computational Linguistics*. Springer, 558–572.

[26] Bingfeng Luo, Yansong Feng, Jianbo Xu, Xiang Zhang, and Dongyan Zhao. 2017. Learning to Predict Charges for Criminal Cases with Legal Basis. In *Proc. of EMNLP*. 2727–2736.

[27] Ejan Mackaay and Pierre Robillard. 1974. *Predicting judicial decisions: The nearest neighbour rule and visual representation of case patterns.* 3(3/4):302–331 pages.

[28] Eneldo Loza Mencia and Johannes Fürnkranz. 2008. Efficient pairwise multilabel classification for large-scale problems in the legal domain. In *Proc. of ECML-PKDD*. Springer, 50–65.

[29] Taro Miyazaki, Kiminobu Makino, Yuka Takei, Hiroki Okamoto, and Jun Goto. 2019. Label Embedding using Hierarchical Structure of Labels for Twitter Classification. In *Proc. of EMNLP*. 6317–6322.

[30] Stuart S Nagel. 1963. Applying correlation analysis to case prediction. *Tex. L. Rev.* 42 (1963), 1006.

[31] Vinod Nair and Geoffrey E. Hinton. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proc. of ICML*. 807–814.

[32] Jinseok Nam, Eneldo Loza Mencía, and Johannes Fürnkranz. 2016. All-in Text: Learning Document, Label, and Word Representations Jointly. In *Proc. of AAAI*. 1948–1954.

[33] Yingwei Pan, Ting Yao, Yehao Li, and Tao Mei. 2020. X-Linear Attention Networks for Image Captioning. In *Proc. of CVPR*. 10968–10977.

[34] Nikolaos Pappas and James Henderson. 2019. GILE: A Generalized Input-Label Embedding for Text Classification. *TACL* 7 (2019), 139–155.

[35] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proc. of NAACL-HLT*. 2227–2237.

[36] Hamed Pirsiavash, Deva Ramanan, and Charless C. Fowlkes. 2009. Bilinear classifiers for visual recognition. In *Proc. of NeurIPS*. 1482–1490.

[37] José A. Rodríguez-Serrano and Florent Perronnin. 2013. Label embedding for text recognition. In *Proc. of BMVC*.

[38] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic Routing Between Capsules. In *Proc. of NeurIPS*. 3856–3866.

[39] Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management* 24, 5 (1988), 513–523.

[40] Octavia-Maria Sulea, Marcos Zampieri, Shervin Malmasi, Mihaela Vela, Liviu P Dinu, and Josef van Genabith. 2017. Exploring the Use of Text Classification in the Legal Domain. In *Proceedings of ASAIL workshop*.

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

[42] Jian Tang, Meng Qu, and Qiaozhu Mei. 2015. PTE: Predictive Text Embedding through Large-scale Heterogeneous Text Networks. In *Proc. of SIGKDD*. 1165–1174.

[43] Joshua B Tenenbaum and William T Freeman. 2000. Separating style and content with bilinear models. *Neural computation* 12, 6 (2000), 1247–1283.

[44] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *JMLR* 9, 11 (2008).

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Proc. of NeurIPS*. 5998–6008.

[46] Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. 2018. Joint Embedding of Words and Labels for Text Classification. In *Proc. of ACL*. 2321–2331.

[47] Lin Xiao, Xin Huang, Boli Chen, and Liping Jing. 2019. Label-Specific Document Representation for Multi-Label Text Classification. In *Proc. of EMNLP*. 466–475.

[48] Nuo Xu, Pinghui Wang, Long Chen, Li Pan, Xiaoyan Wang, and Junzhou Zhao. 2020. Distinguish Confusing Law Articles for Legal Judgment Prediction. In *Proc. of ACL*. 3086–3095.

[49] Wenmian Yang, Weijia Jia, Xiaojie Zhou, and Yutao Luo. 2019. Legal Judgment Prediction via Multi-Perspective Bi-Feedback Network. In *Proc. of IJCAI*. 4085–4091.

[50] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Proc. of NeurIPS*. 5754–5764.

[51] Majid Yazdani and James Henderson. 2015. A Model of Zero-Shot Learning of Spoken Language Understanding. In *Proc. of EMNLP*. 244–249.

[52] Zhou Yu, Jun Yu, Jianping Fan, and Dacheng Tao. 2017. Multi-modal Factorized Bilinear Pooling with Co-attention Learning for Visual Question Answering. In *Proc. of ICCV*. 1839–1848.

[53] Chao Zhang, Zichao Yang, Xiaodong He, and Li Deng. 2020. Multimodal intelligence: Representation learning, information fusion, and applications. *JSTSP* 14, 3 (2020), 478–493.

[54] Honglun Zhang, Liqiang Xiao, Wenqing Chen, Yongkun Wang, and Yaohui Jin. 2018. Multi-Task Label Embedding for Text Classification. In *Proc. of EMNLP*. 4545–4553.

[55] Haoxi Zhong, Zhipeng Guo, Cunchao Tu, Chaojun Xiao, Zhiyuan Liu, and Maosong Sun. 2018. Legal Judgment Prediction via Topological Learning. In *Proc. of EMNLP*. 3540–3549.

[56] Haoxi Zhong, Yuzhong Wang, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. Iteratively Questioning and Answering for Interpretable Legal Judgment Prediction. In *Proc. of AAAI*. 1250–1257.

[57] Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. How Does NLP Benefit Legal System: A Summary of Legal Artificial Intelligence. In *Proc. of ACL*. 5218–5230.