Improving Statute Prediction via Mining Correlations between Statutes

Yi Feng dz1732001@smail.nju.edu.cn
Chuanyi Li lcy@nju.edu.cn
Jidong Ge gjd@nju.edu.cn
Bin Luo luobin@nju.edu.cn
State Key Laboratory for Novel Software Technology, Software Institute, Nanjing University, Nanjing 210093, China

Abstract

The task of statute prediction focuses on determining applicable statutes for legal cases with the inputs of fact descriptions, which is crucial for both legal experts and ordinary people without professional knowledge. Existing works just consider the correspondence from facts to individual statutes and ignore the correlations between statutes. Moreover, charges of cases have associations with statutes. To address these issues, we formulate statute prediction task as a sequence generation problem and propose a novel joint generative model to mine correlations between statutes. By integrating statute prediction task and charge prediction task, we also make model learn associations between statutes and charges. Experiments show our model outperforms several baselines significantly and correlative statutes are predicted accurately.

Keywords: Statute Prediction; Multi-label Classification; Joint Learning

1. Introduction

Statute prediction aims at finding applicable statutes for judging cases with the inputs of case fact descriptions, which is a successful application of Natural Language Processing in the legal field. Statute prediction is highly significant for both legal experts and ordinary people. When faced with numerous complicated cases, legal experts, such as judges and lawyers, have to analyze facts of cases and attend to the opinions put forward by the parties, which are already difficult. And what's worse is searching appropriate statutes from so many candidates. For ordinary people with no professional knowledge, they may spend high expense on legal advice about cases they are involved in. With the convenience of statute prediction, legal experts, as well as ordinary people, can obtain statutes and get legal consult service easily.

There are several challenges for predicting statutes. Because of thousands of different statutes, it is hard to choose applicable ones for cases from a wide range of candidates. Besides, a case may be associated with multiple statutes, which makes statute prediction more complicated. More importantly, statutes are not independent and there are correlations among them. In fact, there are four types of relationships between statutes, namely, inclusive relationship, exclusive relationship, neutral relationship and complementary relationship. Inclusive relationship means violating one statute will inevitably violate another specific one. For example, committing contract fraud is bound to result in guilty of fraud charge and related statutes of them should be cited together for adjudication. On the contrary, theft and robbery are both involved in the loss of property, but if one case is sentenced by theft statutes, robbery statutes is not applicable, which is exclusive relationship. For neutral relation, some laws may or may not co-exist with others. For example, under normal circumstances, being convicted of theft just needs support of theft related statutes and has no direct association with intentional assault related laws. If stealing medicine leads to the death of the victim, additional intentional assault related laws are required. For complementary relation, some statutes complement each other in content. For example, it is necessary to cite justifiable defense regulations and intentional assault related laws when trying to judge justifiable defense cases. Therefore, recommending statutes should not only consider the applicability of individual articles of law, but also take correlations between statutes into account. By capturing these association, inclusive statutes can be predicted together and exclusive ones can be recommended separately. In this paper, we mainly focus on solving the correlation challenge.

Statute prediction is a branch of legal prediction, which has been explored for decades. In the early works, a large amount of rules are designed and employed to detect patterns in texts. When certain conditions are satisfied, correlative results are generated (Kort, 1957; Nagel, 1963; Segal, 1984). But these rules have the limits of giving bias to frequent ones resulting in low accuracy and work poorly in generalization ability. With the successful application of machine learning, some works deal with legal prediction task using classification techniques (Lin et al., 2012; Liu and Chen, 2018; Aletras et al., 2016). Different categorization algorithms are utilized for prediction, such as Rand Forest, Support Vector Machine(SVM). There are also works solving the problem from the text similarity perspective. They find similar cases and recommend statutes referred in them (Liu et al., 2004). Liu et al. (2015) attempt to combine classification and text similarity mechanisms, and unify them into the same architecture so as to exploit both advantages. Though significant improvements have been achieved, the classification-based and text similarity-based methods mainly rely on statistic features and show weakness in capturing semantics. In recent years, neural networks are demonstrated effective in many fields, such as machine translation, dialogue systems, because of advances in mining not only word order relations of sentences but also latent semantics. Researchers propose to employ sophisticated techniques of deep learning to formulate legal prediction (Luo et al., 2017; Hu et al., 2018).

However, the existing works only learn to find correspondence between facts and statutes, and treat statutes as individual labels without consideration of the inner correlations among them. In addition, the charges of cases are also related to statutes. Criminal cases have high probability supported by criminal laws, and divorce cases usually correspond to marriage laws. In real judgment process, judges determine charges of cases first to reduce the scale of candidate statutes. If we can extract these relevance between statutes and charges, it is undoubted to improve statute recommendation task.

To address the above issues, we propose a novel joint generative model(JGM) to capture correlations among statutes, as well as associations between statutes and charges for improving statute prediction. Specifically, we formulate statute recommendation as a sequence generative problem and take advantage of sequence-to-sequence(Seq2Seq) architecture to predict statutes one by one, in which the subsequent results depend on the preceding ones to mine relevance of statutes. Furthermore, we unify charge prediction task and statute recommendation task in the same model by joint learning to make them contribute to each other. With the inputs of case facts, the encoder of our model extract latent semantics of text by self-attention mechanism. Based on the contextual representations, charges of cases are predicted. Then decoder part focuses on the memory of the encoder and statutes that have been predicted to learn to predict applicable statutes in sequence. We evaluate our approach using dataset constructed from real-world cases. Experimental results demonstrate our model offers significant improvements than several baselines. By the advance of mining correlations among statues, relevant statutes are predicted accurately under generative model. Our experiments also show charge prediction provides useful information for improving statute prediction. To summarize, we make the following contributions:

(1) We creatively solve statute correlation challenge and mine correlations among statutes, as well as associations between statutes and charges for improving statute prediction, which is not considered in any existing work.

(2) We formulate statute prediction task as a sequence generation problem and propose a novel joint generative model for predicting correlative statutes. We also unify statute prediction and charge prediction into the same framework jointly for exploiting charge information to assist statute prediction task.

(3) Extensive experiments under real-world datasets demonstrate that our proposed model achieves the best performance compared with several baselines and relevant statutes are predicted accurately. And another improvement can be achieved by incorporating charge information.

2. Related Work

2.1. Legal prediction

Legal prediction focuses on predicting charges, statutes and penalty and so on in the legal domain. In the early works, prediction mainly depends on rule-based methods and numerous rules are defined manually (Kort, 1957; Nagel, 1963; Segal, 1984). With the development of machine learning, many works explore handling the problem under classification framework with the consideration of statistical features. Lin et al. (2012) designed 21 legal factors and extract them for case classification and sentence prediction. Liu and Chen (2018) designed a two-step sentiment analysis method to predict judgment, in which text mining technologies are employed to obtain features from original texts and SVM classifiers are trained to predict judgment category. Aletras et al. (2016) make binary classification for outcome of cases based on N-gram and topic vectors. Besides classification, text similarity-based approaches are also widely used. Liu et al. (2004) define explicit rules to extract features and apply k-Nearest Neighbor(k-NN) algorithm to classify criminal charges. Liu et al. (2015) propose a three-step system for statute prediction, in which classification and text similarity are combined together and show strong performance. In recent years, neural networks are confirmed powerful in capturing semantics. Luo et al. (2017) predict possible relevant statutes to assist charge prediction by recurrent networks with hierarchical attention mechanism. Hu et al. (2018) design several discriminateive attributes for charge prediction. Different from existing works, our model extracts correlations among statutes, as well as associations between statutes and charges for improving statute prediction through joint generation.

2.2. Sequence-to-Sequence Framework

Seq2Seq framework has been used widely for many tasks, especially sequence generation problems, such as machine translation, text summarization. The encoder can transform a variable-length sequence into a fixed-length vector representation, from which the decoder generates a variable-length target sequence. Bahdanau et al. (2015) model machine translation in encoder-decoder system and extend it to learn to align and translate jointly. Zhou et al. (2017) propose a selective gate component to improve encoder-decoder for abstractive sentence summarization. Rush et al. (2015) take advantage of local attention mechanism and generate summary conditioned on the encoder. In this paper, we take advantage of Seq2Seq learning to predict statutes in sequence and capture correlations among them.

2.3. Multi-label Classification

There are multiple statutes applicable to one case, which is a multi-label problem. Existing works mainly take three types of methods to solve multi-label task, i.e., problem transformation, algorithm adaptation and neural networks. Problem transformation strategy converts the original task into binary or multi-class classification. Binary Relevance(BR) (Boutell et al., 2004) trains independent binary classifiers for each label. Classifier Chain(CC) (Read et al., 2011)constructs subsequent classifier with regard to preceding ones' predictions, which is also under binary classification architecture. Label Powerset (LP) (Tsoumakas and Katakis, 2007)deals with multi-label problem as multi-class problem and train classifiers upon all possible label combinations. There are also approaches developing existing algorithms for adapting multi-label task. Clare and King (2001) exploit multi-label entropy to establish decision tree. Zhang and Zhou (2007) modify traditional k-NN technique and utilize priori knowledge for handling multi-label data. In addition to traditional machine learning methods, neural networks are also employed. Zhang and Zhou (2006) propose a pairwise ranking loss function and solve the problem using a fully-connected neural networks. Li et al. (2015) take advantage of joint learning for multi-label text categorization. Different from existing multi-label approaches, we formulate statute prediction task as a sequence generation problem and solve it in using Seq2Seq framework.

3. Approach

In this section, we give detailed description of our generative model based on Seq2Seq learning. First, we describe the definitions of both statute recommendation task and charge prediction task. Then, the two main self-attention based components of model are introduced, i.e., the neural encoder of case facts and the attentive decoder for predicting statutes. At last, we show the output layers and the loss function for training.

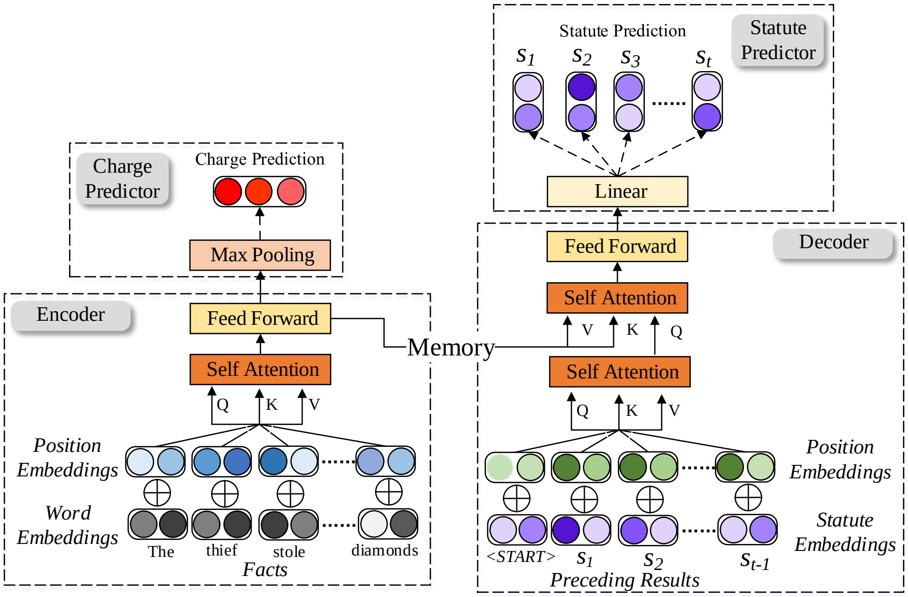

Figure 1: The framework of our proposed JGM.

3.1. Task Formulation

Fact descriptions can be seen as a sequence of words $X = \{x_1, x_2, ..., x_m\}$, where x_m is the input word from a fix-sized vocabulary V. Given fact descriptions of one case, statute prediction aims at finding a set of applicable statutes $\{s_1, s_2, ..., s_t\}$ for adjudicating case, where $s_t \in S$ is an individual statute from the whole set of statutes. Charge prediction takes the same inputs as the statute prediction task and aims at predicting charge of case $c \in C$, where C is charge label set. In this paper, we focus on statute recommendation for Chinese cases.

3.2. Overview

The overview of our model is illustrated in Fig. 1. We also treat statute prediction as a multi-label classification problem. Unlike existing works which consider the mapping relation from texts to labels separately, we propose an encoder-decoder structure to generate statutes step by step. The encoder learns to map the input of fact sequence to a sequence of distributed memory representations which contain latent features among fact texts by self-attention. Based on memory representations, our model predict charges. The decoder predicts an output sequence of statutes one statute at a time. At each generation step, our model utilize the preceding predicted results as history information and the memory representations to focus on fact words attentively.

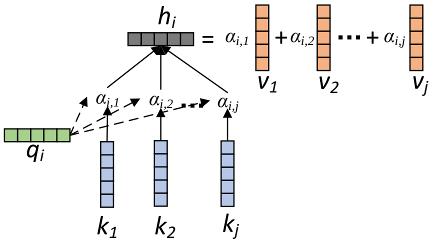

Figure 2: The process of self-attention mechanism.

3.3. Fact Neural Encoder

Given the sequence of fact words $X = \{x_1, x_2, ..., x_m\}$, we first convert x_m into embedding vector $\mathbf{w}_m$ through an embedding matrix $\mathbf{E}_w \in \mathbb{R}^{d_{model} \times |V|}$, where d_{model} is the dimension size of each word and $|V|$ is the size of vocabulary. Then, we can get embedding sequence $X = \{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_m\}$. Inspired by Vaswani et al. (2017), we take advantage of self-attention mechanism to capture semantic information from word embeddings.

The self-attention can be seen as a function that maps a query vector and a set of key vectors with corresponding value vectors to the output. Specifically, the output is computed by a weighted sum of values and the weight of each value is obtained from the vector product of the query and the relative key. Fig. 2 shows the process of self-attention mechanism. Given a query vector $\mathbf{q}_i$ and a set of $(\mathbf{k}_j, \mathbf{v}_j)$ pairs , we calculate as follows:

$$u_{i,j} = (\mathbf{q}_i \odot \mathbf{k}_j) / \sqrt{d_k} \tag{1}$$

$$\alpha_{i,j} = exp\,(u_{i,j}) / \sum_j exp\,(u_{i,j}) \tag{2}$$

$$\mathbf{h}_i = \sum_j \alpha_{i,j} \mathbf{v}_j \tag{3}$$

where $\mathbf{q}_i$ and $\mathbf{k}_j$ have the dimension of d_k, and $\mathbf{v}_j$ has the dimension of d_v. $\odot$ denotes the dot product. $\mathbf{h}_i$ is the output attentive information for the current $\mathbf{q}_i$. Because self-attention can be operated in parallel, we compute attentive information for queries simultaneously:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{4}$$

where Q is the set of queries, and K, V are sets of keys and values separately.

Because there are no recurrent or convolutional components in the self-attention mechanism, we need to incorporate extra information that reflects positional relationships between words:

$$PE(pos, dim) = \begin{cases} sin(pos/10000^{dim/d_{model}}) & if\ dim\ is\ even \\ cos(pos/10000^{dim-1/d_{model}}) & if\ dim\ is\ odd \end{cases} \tag{5}$$

where *pos* is the position and *dim* is the dimension index. If the dimension index is even, we use a sine function. If it is odd, we use cosine. The position embedding has the same dimension size as word embedding and we sum them as the final fact representations:

$$\mathbf{e}_m = \mathbf{w}_m \oplus \mathbf{p}_m \tag{6}$$

where $\mathbf{p}_m$ is the position embedding and $\oplus$ is element-by-element sum.

In our encoder structure, we apply self-attention to each $\mathbf{e}_m$. The Q, K, V are all $E = \{\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_m\}$:

$$\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_m = \mathit{Attention}(E, E, E) \tag{7}$$

After obtaining attention information $\{\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_m\}$, we input them into a position-wise full connected feed-forward network:

$$\mathbf{m}_1, \mathbf{m}_2, ..., \mathbf{m}_m = Feedforward(\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_m) \tag{8}$$

where $\mathbf{m}_m$ is the memory information generated by the encoder, which contains latent features of input texts and further utilized by the decoder.For charge prediction, we apply max-pooling to memory vectors and learn charge probabilities by multi-class classification:

$$\hat{\mathbf{c}} = f(maxpooling\,\{\mathbf{m}_1, \mathbf{m}_2, ..., \mathbf{m}_m\}) \tag{9}$$

where $\hat{\mathbf{c}}$ is the prediction probability distribution over all charge categories. f is the multi-class classification function.

3.4. Attentive Decoder

To consider correlations among statutes, the decoder generates statutes in sequence and enable model to depend on known recommended statutes. The decoder structure also takes advantage of self-attention to predict statutes. But different from the encoder, the decoder focuses on two kinds of information, i.e., the preceding generated statutes and the memory from the encoder. Thus, two layers of self-attention is used. When generating the output statute s_t of timestamp t, the decoder firstly calculates self-attention over all previous statutes $s_1, s_2, ..., s_{t-1}$ like the encoder. Each s_i is embedded into $\mathbf{o}_i$ by a matrix $\mathbf{E}_o \in \mathbb{R}^{d_{model} \times |S|}$. The same position embeddings are also employed to work as positional features:

$$\mathbf{z}_i = \mathbf{o}_i \oplus \mathbf{p}_i \tag{10}$$

where $\mathbf{z}_i$ is the element-by-element sum of statute embedding and position embedding. Then we can get the final inputs $Z = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_{t-1}\}$ of decoder. The first layer of attention in decoder is computed as:

$$\mathbf{l}_1, \mathbf{l}_2, ..., \mathbf{l}_{t-1} = \mathit{Attention}(Z, Z, Z) \tag{11}$$

Because not all fact words make the same contribution when generating the current statute, we should focus on them attentively too. After self-attention upon $s_1, s_2, ..., s_{t-1}$, we conduct the encoder-decoder attention over memory information. The Q is the attention information $L = \{\mathbf{l}_1, \mathbf{l}_2, ..., \mathbf{l}_{t-1}\}$, and the K, V are $M = \{\mathbf{m}_1, \mathbf{m}_2, ..., \mathbf{m}_m\}$ from the encoder:

$$\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_{t-1} = \mathit{Attention}(Z, M, M) \tag{12}$$

This enables each position in the decoder to attend over all positions in the fact sequence. Then, we apply a position-wise full connected feed-forward network to filter the encoder-decoder attention results:

$$\mathbf{g}_1, \mathbf{g}_2, ..., \mathbf{g}_{t-1} = Feedforward(\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_{t-1}) \tag{13}$$

Finally, a linear projection layer that is shared with statutes embedding matrix $\mathbf{E}_o$ is used to map feed-forward results to the last statute probability prediction distribution:

$$\hat{\mathbf{y}}_t = Linear(\mathbf{g}_{t-1}, \mathbf{E}_o) \tag{14}$$

where $\mathbf{y}_t$ is prediction probability distribution over all statute candidates and the maximum one is taken as the predicted statute s_t.

3.5. Training

For statute prediction, the training object is to minimize cross-entropy over all timestamps:

$$\pounds_s = -\sum\sum \mathbf{y}_t log \hat{\mathbf{y}}_t \tag{15}$$

where $\hat{\mathbf{y}}_t$ is the predicted probability distribution of statutes and $\mathbf{y}_t$ is the ground-truth. We sum all timestamps and all training samples. For charge prediction, we also take cross-entropy as the loss function:

$$\pounds_c = -\sum \mathbf{c} log \hat{\mathbf{c}} \tag{16}$$

where $\hat{\mathbf{c}}$ is the probability distribution of charges and $\mathbf{c}$ is the ground-truth. We sum all training samples. The overall loss is the sum of the two tasks:

$$\pounds = \pounds_s + \lambda \pounds_c \tag{17}$$

where λ is the weight of charge prediction task.

4. Experiment

In this section, we evaluate our model on the datasets constructed from real-world cases. We first introduce our datasets in detail. Then, we give description about the hyper-parameters of our model, as well as evaluation metrics. Finally, we provide the analysis of experimental results.

4.1. Dataset

To verify the effectiveness of our proposed method under real condition, we collect a large number of real-world cases as our dataset from China Judgments Online[1]. We split them into 80% for training, 10% for validation and 10% for test. The details of our dataset are shown in Table 1.

[1] http://wenshu.court.gov.cn/

Table 1: The statistic of the dataset. #Words/Case denotes the average number of words per case. #Statutes/Case is the average number of statutes cited by case.

Case	**Vocabulary**	**Charge**	**Statute**	**#Words/Case**	**#Statutes/Case**
82756	259478	13	226	263	4.3

4.2. Experiment Settings

We employ Jieba [2] to segment the facts of cases, because there is no space between Chinese words. When generating the vocabulary, we neglect low frequency words with counts less than 30 and the final vocabulary list size is 14298, i.e., $|V|$=14298. The unknown words are represented by token '<UNK >'. In the period of predicting statutes, we input special token '<START>' to mark the beginning of generating. The greedy strategy is adopted to select statutes at each timestamps. When special token '</END>' is output, we stop the generation process. We set the embedding size to 128, i.e., the d_{model} is 128. For training, we make use of Adam (Kingma and Ba, 2015) to optimize the loss function with β_1= 0.9, β_2= 0.98 and ϵ= 10-9. The batch size is set to 64 and the epoch is 20. Besides, we apply residual connection(He et al., 2016) for each self-attention layer and feed-ward layer. Before predicting statutes and charges, layer normalization(Ba et al., 2016) is employed as regularization. All experiments are repeat for 5 times and average results are taken. We evaluate our model using precision, recall and F1 values per case denoted as P_C, R_C, $F1_C$ and per statute category denoted as P_{Mi}, R_{Mi}, $F1_{Mi}$.

4.3. Baselines

TPP: Liu et al. (2015) propose a three-step framework for statute prediction, in which SVM-based multi-label classifier combined with text similarity are utilized and prominent results are demonstrated in real cases. We take it as a strong baseline.

SE: Luo et al. (2017) also treat statute prediction as a multi-label classification and statutes are seen as labels. Specifically, they utilize bag-of-word TF-IDF features and train several binary SVM classifiers for each label.

LP(Label Powerset): Tsoumakas and Katakis (2007) transform multi-label as a multi-class problem. Each combination of labels is seen as an individual category. We utilize multi-class SVM as classifier.

DT: Clare and King (2001) exploit multi-label entropy to establish decision tree for multi-label task. They implement it in the algorithm adaptation way.

ML-KNN: Zhang and Zhou (2007) modify k-NN to adapt to multi-label classification. Different from classification-based methods, ML-KNN take similarity between texts into account.

CNN: Kim (2014) employ Convolutional Neural Networks(CNN) for sentence-level classification, which shows advances on several tasks.

[2] https://pypi.org/project/jieba/

Table 2: The values of metrics achieved by JGM compared with several baselines. For convenience, we denote our whole model without charge prediction as JGM and '+Charge Prediction' denotes the whole model.

Models	$\mathbf{P_C}$	$\mathbf{R_C}$	$\mathbf{F1_C}$	$\mathbf{P_{Mi}}$	$\mathbf{R_{Mi}}$	$\mathbf{F1_{Mi}}$
TPP(Liu et al., 2015)	0.799	0.705	0.749	0.794	0.650	0.715
SE(Luo et al., 2017)	0.752	0.695	0.722	0.745	0.637	0.687
LP(Tsoumakas and Katakis,2006)	0.788	0.693	0.737	0.774	0.635	0.698
DT(Clare and King, 2001)	0.579	0.588	0.584	0.546	0.559	0.552
ML-KNN(Zhang and Zhou, 2007)	0.781	0.737	0.758	0.767	0.701	0.733
CNN(Kim, 2014)	0.704	0.779	0.740	0.696	0.732	0.714
Bi-GRU+ATT+Joint	0.738	0.766	0.752	0.725	0.716	0.721
LSTM-based Seq2Seq	0.814	0.770	0.791	0.795	0.741	0.767
JGM	**0.819**	**0.797**	**0.808**	**0.800**	**0.770**	**0.785**
+Charge Prediction	**0.824**	**0.798**	**0.811**	**0.803**	**0.770**	**0.786**

Bi-GRU+ATT+Joint: To illustrate the effectiveness of related task, i.e., charge prediction, we also compare our model with a neural networks joint model, in which statute recommendation and charge prediction are solved in a same classification architecture. We utilize Bi-GRU to extract semantic, and two full-connected neural networks predict statutes and charges separately. We apply an attention component to focus on key words.

LSTM-based Seq2Seq: Sequence generative models that can generate objects sequentially are widely used in many tasks, such as machine translation, text summarization. These models mainly employ LSTM as the encoder and the decoder. LSTM has ability to extract order information and model long short-term dependence.

4.4. Experimental Results

The results compared with baselines are shown in Table 2. The proposed model achieves better performance on metrics than all baselines, which demonstrates the effectiveness of our model. Specifically, compared with existing statute prediction works, our model outperforms TPP (Liu et al., 2015)in $F1_C$ by 6.2% and $F1_{Mi}$ by 7.1% , as well as SE (Luo et al., 2017) in $F1_C$ by 8.9% and $F1_{Mi}$ by 9.9% . The statistics models suffer from the ignorance of semantics among texts and only split the feature space to make classification. Our model can understand texts from the perspective of linguistic and make full use of semantic information. TPP, SE are based on SVM that has bias to frequent statutes, which results in high precision. Compared with them, the proposed model is more balanced. Our model also performs better than other classifier-based multi-label methods, i.e., LP and DT. Though ML-KNN makes use of text distance to find similar cases for statute prediction and perform well than other multi-label methods, it needs much more time to retrieve statutes that is not applicable in real scenario and performs less effective than JGM. The experimental results also convince that our model offers significant improvements over deep learning based frameworks. Compared with CNN, LSTM-based Seq2Seq takes advantage of

Table 3: Comparisons with TPP in several sub-datasets with different degrees of statute dependence. #Case denotes the number of cases. #statute is the number of statutes.

#Case	**#Statute**	**$\mathbf{F1_C}$**			**$\mathbf{F1_{Mi}}$**		
		TPP	**JGM**	**Improvement**	**TPP**	**JGM**	**Improvement**
20401	86	0.75	0.787	4.90%	0.704	0.767	8.90%
40660	123	0.764	0.813	6.40%	0.723	0.799	10.50%
57517	168	0.732	0.805	10.00%	0.691	0.789	14.20%
82756	226	0.749	0.811	8.10%	0.715	0.786	9.90%

generative structure that extracts correlations among statutes and thus achieves prominent improvements. We can also observe that the Bi-GRU+ATT+Joint model achieves better results than CNN because of learning the associations between statutes and charges, which confirms the advantages of charge information. By extracting relevance among statutes, as well as integrating charge prediction task, JGM is superior to baselines in large margin. The accuracy of charge prediction by JGM can reach 98.2%. Even without learning charge information jointly, our model is still able to get the best recall and F1 values.

To further illustrate the advantage of taking correlations among statutes into account, we split the dataset into several sub-datasets and test the ability of our model in handling complicated statute dependence. As shown in Table 3, the number of statutes indicates complexity of dependence among statutes. Our model shows advances over TPP in large margin, especially in complicated conditions. With the increase of number of statutes, our model gets better improvements than TPP. However, in the most complicated condition, the improvement has growth limit. It may be because the latent relevance between statutes are too difficult for JGM to learn when there are a large amount of statues, but our model is still more effective than TPP.

4.5. Correlation Analysis

To explore how JGM can outperform TPP in Table 3 with the increase of complicated dependence. We select several representative samples for illustrating the importance of capturing correlations among statutes in our model, which is shown in Table 4. We compare our model and basic TPP in predicting relevant statutes. In the first sample that is a traffic accident case, the law I6, I7 are inclusive. If I6 is applicable, I7 is applicable too. Moreover, in the theft sample, C38 and C39 are two subtle different statutes for judging theft and robbery separately, which means they are mutually exclusive and can't be cited together. TPP wrongly predict C38 and C39 at the same time. Besides, in the second sample, D11 and D12 are often cited together and the confidence from D11 to D12 is about 0.864, which means D12 has 86.4% chance of being cited if D11 occurs. Compared with TPP, JGM can predict these relevant statutes accurately. Inclusive statutes can be predicted together and exclusive ones can be recommended separately.

Table 4: Representative samples of predicted results of our model and TPP. The red statutes means they are inclusive. The blue statutes indicate they are exclusive. The orange statutes mean they are usually cited together for judgment.

Case	Citations	TPP	JGM
Traffic Accident	I3, I12, I6, I7	I12, I3, I9, I6	I3, I6, I7
Dangerous Driving	I3, D11, D12, D9	I3, D11	I3, D11, D12, D10
Theft	C38, C33, C30, C32, C34, C35, C40, C1	C38, 39, C33, C34, C40, C1	C38, C33, C30, C32, C34, C35

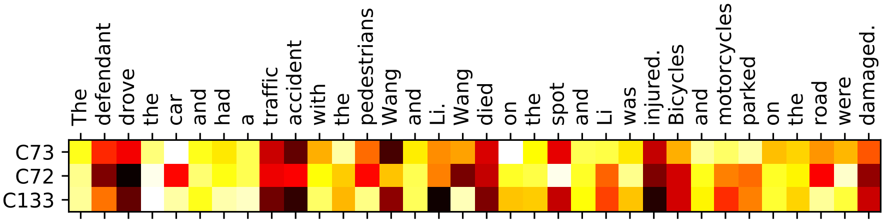

Figure 3: The visualization of self-attention mechanism.

The self-attention mechanism is used in two different ways by our model. One is input attention, which help the each position of encoder and decoder attend to all other positions. Another is the attention from the encoder to the decoder, which allows each position of the decoder to focus on fact words discriminately and give the important words large weights. We exhibit the attention mechanism from encoder to decoder of one sample in Fig. 3. There are three articles of law relevant to this sample. The darker color means the more importance of fact words when generating the current timestamp statute. From the Fig. 3, we can see the words having abundant semantic information play key roles when generating statutes.

4.6. Error Analysis

Our model achieves best performance over all baselines. However, it still has great potential for making further progress. The errors of our proposed model are raised mainly by the following three causes: (1)Data Imbalance: There exists data imbalance of statutes in our dataset. The frequencies of labels diversify extremely. For the most frequent one, it occurs in 63.6% of all samples. Meanwhile, there are lots of labels only being cited by dozens of times. It is difficult to learn the correspondence from text features to these statutes with low frequencies. From Fig. 4(a), we can see high frequency ones(over 5000) can obtain 0.849 in F1, in sharp contrast with low frequency ones(10-100) whose F1 scores nearly equal to 0. (2) Length of texts and statutes: the fact descriptions provide all input information

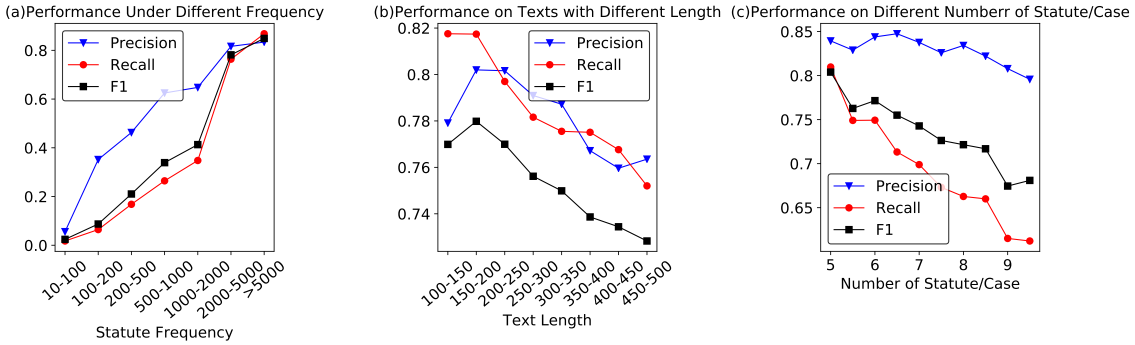

Figure 4: The error analysis of JGM.

and the features contained in them contribute to the last results directly. We test the relevance between the length of fact description and the prediction performance. As show in Fig. 4(b), metrics rise slightly at first and then fall. It indicates that more features contribute to prediction result, but too long texts make it difficult for model to extract semantics in them. (3) Number of statutes: Moreover, we test the performance of our model under different number of cited statutes per case. Fig. 4(c) shows that more statutes make our model perform poorly. The cases citing too many statutes are complicated and it is difficult to predict all applicable statutes.

5. Conclusion

In this paper, we propose a novel joint generative model for statute prediction. Through sequence generation framework, the correlations among statute can be extracted for improving statute prediction, which is not considered in any existing work. Besides, we integrate statute recommendation and charge prediction to mining the associations between statutes and charges. The experiments demonstrate the significant advances of our proposed model and taking correlations among statutes into account indeed contributes to statute prediction: (1)Compared with the strong baseline TPP, our model increase in $F1_C$ by 6.2% and $F1_{Mi}$ by 7.1%; (2)Another improvement can be achieved when incorporating charge prediction task; (3) Correlation analysis shows relevant statutes can be predicted accurately by our model. Though promising, our model only obtains the embeddings of statutes in the training process. In the future work, we will explore how to utilize the texts of statutes directly for enhancing our model.

6. Acknowledgements

This work was supported by the National Key R&D Program of China (2016YFC0800803), the National Natural Science Foundation of China (No. 61802167), and the Fundamental Research Funds for the Central Universities. Chuanyi Li and Jidong Ge are both the corresponding authors.

References

Nikolaos Aletras, Dimitrios Tsarapatsanis, Daniel Preotiuc-Pietro, and Vasileios Lampos. Predicting judicial decisions of the european court of human rights: a natural language processing perspective. *PeerJ Computer Science*, 2:e93, 2016. doi: 10.7717/peerj-cs.93. URL https://doi.org/10.7717/peerj-cs.93.

Lei Jimmy Ba, Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016. URL http://arxiv.org/abs/1607.06450.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.0473.

Matthew R. Boutell, Jiebo Luo, Xipeng Shen, and Christopher M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, 2004. doi: 10.1016/j.patcog.2004.03.009. URL https://doi.org/10.1016/j.patcog.2004.03.009.

Amanda Clare and Ross D. King. Knowledge discovery in multi-label phenotype data. In Luc De Raedt and Arno Siebes, editors, *Principles of Data Mining and Knowledge Discovery, 5th European Conference, PKDD 2001, Freiburg, Germany, September 3-5, 2001, Proceedings*, volume 2168 of *Lecture Notes in Computer Science*, pages 42–53. Springer, 2001. ISBN 3-540-42534-9. doi: 10.1007/3-540-44794-6_4. URL https://doi.org/10.1007/3-540-44794-6_4.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016. ISBN 978-1-4673-8851-1. doi: 10.1109/CVPR.2016.90. URL https://doi.org/10.1109/CVPR.2016.90.

Zikun Hu, Xiang Li, Cunchao Tu, Zhiyuan Liu, and Maosong Sun. Few-shot charge prediction with discriminative legal attributes. In Emily M. Bender, Leon Derczynski, and Pierre Isabelle, editors, *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 487–498. Association for Computational Linguistics, 2018. ISBN 978-1-948087-50-6. URL https://aclanthology.info/papers/C18-1041/c18-1041.

Yoon Kim. Convolutional neural networks for sentence classification. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751. ACL, 2014. ISBN 978-1-937284-96-1. URL http://aclweb.org/anthology/D/D14/D14-1181.pdf.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA,* *USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Fred Kort. Predicting supreme court decisions mathematically: A quantitative analysis of the "right to counsel" cases. *American Political Science Review*, 51(1):1–12, 1957.

Li Li, Houfeng Wang, Xu Sun, Baobao Chang, Shi Zhao, and Lei Sha. Multi-label text categorization with joint learning predictions-as-features method. In Màrquez et al. (2015), pages 835–839. ISBN 978-1-941643-32-7. URL http://aclweb.org/anthology/D/D15/D15-1099.pdf.

Wan-Chen Lin, Tsung-Ting Kuo, Tung-Jia Chang, Chueh-An Yen, Chao-Ju Chen, and Shou-de Lin. Exploiting machine learning models for chinese legal documents labeling, case classification, and sentencing prediction. *IJCLCLP*, 17(4), 2012. URL http://www.aclclp.org.tw/clclp/v17n4/v17n4a4.pdf.

Chao-Lin Liu, Cheng-Tsung Chang, and Jim-How Ho. Case instance generation and refinement for case-based criminal summary judgments in chinese. *J. Inf. Sci. Eng.*, 20(4):783–800, 2004. URL http://www.iis.sinica.edu.tw/page/jise/2004/200407_12.html.

Yi-Hung Liu and Yen-Liang Chen. A two-phase sentiment analysis approach for judgement prediction. *J. Inf. Sci.*, 44(5):594–607, 2018. doi: 10.1177/0165551517722741. URL https://doi.org/10.1177/0165551517722741.

Yi-Hung Liu, Yen-Liang Chen, and Wu-Liang Ho. Predicting associated statutes for legal problems. *Inf. Process. Manage.*, 51(1):194–211, 2015. doi: 10.1016/j.ipm.2014.07.003. URL https://doi.org/10.1016/j.ipm.2014.07.003.

Bingfeng Luo, Yansong Feng, Jianbo Xu, Xiang Zhang, and Dongyan Zhao. Learning to predict charges for criminal cases with legal basis. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2727–2736. Association for Computational Linguistics, 2017. ISBN 978-1-945626-83-8. URL https://aclanthology.info/papers/D17-1289/d17-1289.

Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton, editors. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, 2015. The Association for Computational Linguistics. ISBN 978-1-941643-32-7. URL http://aclweb.org/anthology/D/D15/.

Stuart S. Nagel. Applying correlation analysis to case prediction. *Texas Law Review*, 42:1006, 1963.

Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine Learning*, 85(3):333–359, 2011. doi: 10.1007/s10994-011-5256-5. URL https://doi.org/10.1007/s10994-011-5256-5.

Alexander M. Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In Màrquez et al. (2015), pages 379–389. ISBN 978-1-941643-32-7. URL http://aclweb.org/anthology/D/D15/D15-1044.pdf.

Jeffrey A. Segal. Predicting supreme court cases probabilistically: The search and seizure cases, 1962–1981. *American Political Science Review*, 78(4):891–900, 1984.

Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *IJDWM*, 3(3):1–13, 2007. doi: 10.4018/jdwm.2007070101. URL https://doi.org/10.4018/jdwm.2007070101.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6000–6010, 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.

Min-Ling Zhang and Zhi-Hua Zhou. Multi-label neural networks with applications to functional genomics and text categorization. *IEEE Trans. Knowl. Data Eng.*, 18(10):1338–1351, 2006. doi: 10.1109/TKDE.2006.162. URL https://doi.org/10.1109/TKDE.2006.162.

Min-Ling Zhang and Zhi-Hua Zhou. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, 2007. doi: 10.1016/j.patcog.2006.12.019. URL https://doi.org/10.1016/j.patcog.2006.12.019.

Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. Selective encoding for abstractive sentence summarization. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1095–1104. Association for Computational Linguistics, 2017. ISBN 978-1-945626-75-3. doi: 10.18653/v1/P17-1101. URL https://doi.org/10.18653/v1/P17-1101.