Predicting Appropriate Charges of Chinese Criminal Cases Based on Fact Descriptions

Abstract

In this paper, we investigate the problem of determining the appropriate charges for a certain case under the circumstance of statutory law. This problem is important when users communicate with legal assistance systems with natural language, since we need to first determine the category of the case before giving any useful responses. We propose that judgement documents of statutory law countries like China provides natural training data for this task and we find that model multi-label document classification technics fit this problem well. Furthermore, we also find that using additional information of law articles can further improve the model performance in the system of statutory law. TODO: polish the abstract

1 Introduction

Determining appropriate charges (like *fraud*, *theft*, and *homicide*) of a case is helpful for legal assistance systems when the input is natural language. This is especially important when the user has no idea of the legal background of a case, where the only input he (or she) can give is the fact description of the case. For example, if the user interface is dialogue system, one can first determine suitable charges of the current case based on the user description, and then use this to guide the subsequent conversation. If the user wants to find similar cases, one can use the predicted charges of the query case to filter out irrelevant results. Furthermore, if one wants to build an end-to-end judgement suggestion system, where the input is the description of a case, he (or she) also needs to identify appropriate charges before suggesting corresponding penaties.

All these situations, if the user does not point out the charges directly, require us to predict appropriate charges of a case based solely on the fact descriptions. However, this fact-based charge prediction task is not easy: (1) Multiple crimes may be involved in a single case, which means we need to conduct charge prediction in the multi-label classification paradigm. (2) The differences between two charges can sometimes be subtle. For example, in the context criminal cases in China, distinguishing *intentional homicide* from *negligent homicide* involves detailed analysis of the behavior of the defendant, and *acceptance of bribes* differs from *acceptance of bribes by a non-state functionary* in the occupation of the defendant. (3) Although we can expect the model to implicitly learn the legal background of the judgement through massive training data, the charge prediction is still not convincing enough if no law articles are involved in the prediction. This problem is prominent in countries using civil law system, e.g. China (except Hong Kong), where the judgement is made only based on statutory law. Even in countries using common law system, e.g. the United States (except Louisiana), where the judgement is based mainly on decisions of previous cases, there are still some statutory laws that need to be followed when making judgements.

Therefore, to solve this fact-based charge prediction task, we need a multi-label classification model, that can effectively capture the overall framework along with important details of the fact description, and is able to extract and utilize relevant law articles to build the bridge from the fact description to appropriate charges as well.

Our fact-based charge prediction task is closely related to the thread of work on predicting the results of a case since the judgement of a case often involves deciding appropriate charges. Previous works on this thread mainly consider a binary classification paradigm. The target is either to decide whether the outcome will side with the plaintiff or defendant [Aletras *et al.*, 2016], or will the present court affirm or reverse the decision of a lower court [Katz *et al.*, 2016] [1]. Instead of case level prediction, some researches also focus on predicting the votes of each justice [Martin and Quinn, 2002; Lauderdale and Clark, 2014; Sim *et al.*, 2015]. Despite their binary prediction nature, these methods either do not use fact descriptions or just capture shallow semantic meaning of the facts, e.g. using bag-of-words. Furthermore, none of these works employ relevant law articles during prediction. Therefore, these methods are not suitable for our task.

In this paper, we focus specifically on predicting the charges of criminal cases in China, which began to officially publish the judgement documents on China Judgements Online[2] since 2013. Although these judgement documents are

[1] [Katz *et al.*, 2016] also use an additional *other* class to represent other complex outcomes.

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

unstructured, we can use rules and regular expressions to extract the facts description, relevant law articles, and final charges of the case. This naturally provides us with a high-quality large-scale training data set for our task.

To make conprehensive understanding of the fact description, we propose to use the framework of the Hierarchical Attention Network (HAN) [Yang *et al.*, 2016] for document embedding. Specifically, we use a sentence level and a document level Gated Recurrent Unit (GRU) to embed each word and each sentence along with their contexts. Then we use attention mechanism to select the most informative words or sentences for sentence and document embedding respectively. To handle the multi-label nature of the problem, we convert the multi-label target to label distribution, and then use cross entropy as loss function. We find this method works well in our experiments and significantly outperforms the baseline bag-of-words method.

To get support from law articles, we first use a simple bag-of-words-based article classifier to quickly filter out most irrelevant articles. Then we attentively aggregate the retained top k articles for further classification with an additional attentive aggregation module. Although the top k articles are noisy, the experimental results show that our attentive aggregation module can further attend to relevant ones and thus improve the prediction performance.

This work offers an effective way to predict appropriate charges of a case based solely on the fact descriptions. Our main contributions are: (1) We propose that the public judgement documents provides natural high-quality large-scale training data for tasks like fact-based charge prediction and relevant article extraction. (2) We propose a neural network model that can jointly utilize the case facts and the extracted relevant articles to for charge prediction. (3) By evaluating our model on human labeled news data, we show that the model trained on judgement documents have reasonable generalization ability on the text written by people who are not legal practitioners. TODO: polish this paragraph

2 Related Work

Our work is closely related to document classification, which is one of the oldest tasks in natural language processing. The most classic method is to combine bag-of-words features with varies classifiers [Joachims, 1998]. Recently, neural network models like Convolutional Neural Network (CNN) [Kim, 2014] have been applied to document classification and achieve good performance. [Tang *et al.*, 2015] proposes a two-layer scheme, where they use recurrent neural network (RNN) or CNN for sentence embedding, and another RNN for document embedding. [Yang *et al.*, 2016] further adds attention mechanism to the two-layer scheme to distinguish important words or sentences from unimportant ones. As for multi-label document classification, two loss functions are commonly used. The first one is binary cross entropy [Nam *et al.*, 2014], which treats the multi-label classification task as multiple binary classification tasks. The second one is cross entropy [Kurata *et al.*, 2016]. In training phase, it converts the multi-label target to label distribution. After that, it uses the validation set to select a threshold, and consider all the classes with scores higher than the threshold to be positive. In our experiment, we find the latter one converges faster and performs better, so we will use the latter one in this paper.

Since the charge of the defendant is often part of the judgement result of a case, our task is also closely related to the thread of work on predicting the outcome of a case. Some work focuses on predicting wether the plaintiff will win or not [Aletras *et al.*, 2016], while others try to predict whether the present court will affirm or reverse the decision of a lower court [Katz *et al.*, 2016]. The method can be domain specific logical model [Bruninghaus and Ashley, 2003], multi-layer perceptron [Bench-Capon, 1993], SVM [Aletras *et al.*, 2016] and random forest [Katz *et al.*, 2016]. Instead of predicting overall binary result, our task focuses on the detailed result of the case and our output may contain multiple charges.

Another related thread of work is predicting relevant law articles based on case facts. This task aims to find relevant law articles that can be applied to the given case. [Liu *et al.*, 2015] proposes to first use the SVM model for basic article classification, and then use some reranking methods to get the final relevant article list. In our model, however, since we use attention mechanism on the extracted articles, we only care the recall@k rather than the ranking quality in the relevant article extraction step. Therefore, we will simply use SVM for relevant article extraction in our experiment.

Our work also shares the same spirit with the legal question answering task that relevant law articles are import for decisions in civil law system. This task is proposed by the Competition on Legal Information Extraction/Entailment (COLIEE) in 2014[3], and it aims at answer the yes/not questions in the in Japanese legal bar exams. The task first requires participants to extract relevant Japanese Civil Code articles, and then participants will use these articles to answer the question. The article extraction phase is often treated as an information retrieval task, and methods like tf-idf, LDA, and rank-SVM have been applied [Kim *et al.*, 2014]. The question answering phase is often treated as a textual entailment task and methods like CNN [Kim *et al.*, 2015] have been used. There is another thread of work that tries to answer the multiple-choice questions in the USA National Bar Exam [Fawei *et al.*, 2016; Adebayo *et al.*, 2016]. Since the United States operates on common law system, relevant article extraction phase is not employed in these works.

3 Data Preparation

Our data come from the judgement documents published on China Judgements Online. The Chinese government has been publishing the judgement documents on it since 2013[4]. We randomly choose 50,000 judgements as training data, 5,000 for validation and 5,000 for test. To ensure enough training data for each charge, we keep the charges that appear more than 50 times in our training data. As for law articles, we only keep articles in Chinese Criminal Law. In our final dataset, there are 50 distinct charges and 321 distinct articles. About 3.6% cases contain more than one charges, and 94.2% cases

[3] `http://webdocs.cs.ualberta.ca/~miyoung2/jurisin_task/index.html`

[4] There are also judgements before 2013.

contain more than one law articles. The fact description part of each judgement document contains 383 words, and 14 sentences on average.

One of the judgements is shown in Figure 1. Although there does not exist a strict rule for formatting a judgement document, we can still discover some patterns in it. A typical judgement document often starts with a brief description of the procedure followed before the judgement, from the start of the prosecution until the case is decided. The procedure is often followed by the facts of the case. After that, the court will conclude the case and provide relevant law articles that can be applied to the case. Finally, the sentence part will list the charges of the defendant along with corresponding penalties.

We find that the fact description part often starts with the clause 经审理查明 (after hearing, our court identified that), and the court view part often starts with the clause 本院认为 (our court hold that). Therefore we extract the texts between these two clauses as fact description. Since the mentions of charges do not have many variations in judgement documents, we manually build a list that contains the possible variations of each charge based on a public criminal charge list[5]. Then the list is used to find accusation mentions in the sentence part with exact matching. As for law articles, since the article is often mentioned in a fixed patterns, we simply use regular expressions to identify the article mentions.

Note that we only keep the cases with only one defendant. We discard cases with multiple defendants because it is hard to separately relate each defendant to his (or her) corresponding facts, articles and charges due to the unstructured nature of the judgement document.

4 Approach

TODO: re-examine the equations and annotations The framework of our fact-based charge prediction model is shown in Figure 2. First, the input fact description is passed to a document embedding module. Concurrently, the input fact description is also passed to a relevant article extraction module to find top k relevant law articles. Each article is also passed to a document embedding module, and the article embeddings are further passed to a sequence encoder to get the final embedding of each article. The fact embedding generated previously is used to generate the attention over the extracted articles. The article embeddings are then aggregated with the attention distribution, and fed to the softmax function along with the fact embedding to predict the charge distribution of this case.

4.1 Document Embedding

Our document embedding module is based on the HAN model proposed by [Yang *et al.*, 2016]. The framework is shown in Figure 3.

Sequence Encoder A sentence is composed of a sequence of words, and a document is composed of a sequence of sentences. It is important to consider the correlation of the elements when encoding a sequence. Here we use the bi-directional GRU model for sequence encoding, which is an extension of the GRU model [Cho *et al.*, 2014] that use a forward and a backward GRU to model the sequence separately, and then combine the results of both GRUs. Given a sequence $[x_1, x_2, ..., x_T]$ where x_t is the embedding of the element at position t, the encoding result is:

$$h_{ft} = GRU_f(x_t) \tag{1}$$

$$h_{bt} = GRU_b(x_t) \tag{2}$$

$$h_t = [h_{ft}, h_{bt}] \tag{3}$$

where GRU_f is the forward GRU that processes the sequence from left to right, and GRU_b is the backward GRU that processes the sequence from right to left. h_t is the state of the bi-directional GRU model at time t, which is generated by concatenating the states of the two GRU models at time t. The state of a single GRU model at time t is calculated by:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{4}$$

with a little abuse of annotations, h_t here stands for the state of time t of a single GRU model. $z_t \in [0, 1]$ is the update gate, h_{t-1} is the previous state, $\tilde{h}_t$ is the candidate state and $\odot$ is element wise product. The update gate that controls the interpolation of the previous state and the candidate state is calculated by:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{5}$$

where x_t is the input at time t, W_z and U_z are weight matrices, b_z is the bias and σ is the sigmoid function. The candidate state $\tilde{h}_t$ is calculated by:

$$\tilde{h}_t = tanh(W_h x_t + r_t \odot U_h h_{t-1} + b_h) \tag{6}$$

where r_t is the reset gate, which is controls how information does the previous state contributes to the candidate state, and it is calculated by:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{7}$$

Element Attention Since not all elements of a sequence contributes equally to the final output, the attention mechanism is used to select the most import parts of a sequence. Given the GRU output $[h_1, h_2, ..., h_T]$ of a sequence, we calculate a sequence of attention values $[\alpha_1, \alpha_2, ..., \alpha_T]$ where $\alpha_t \in [0, 1]$ and $\sum_t \alpha_t = 1$. The final embedding of the sequence is calculated as:

$$h = \sum_{t=1}^{T} \alpha_t h_t \tag{8}$$

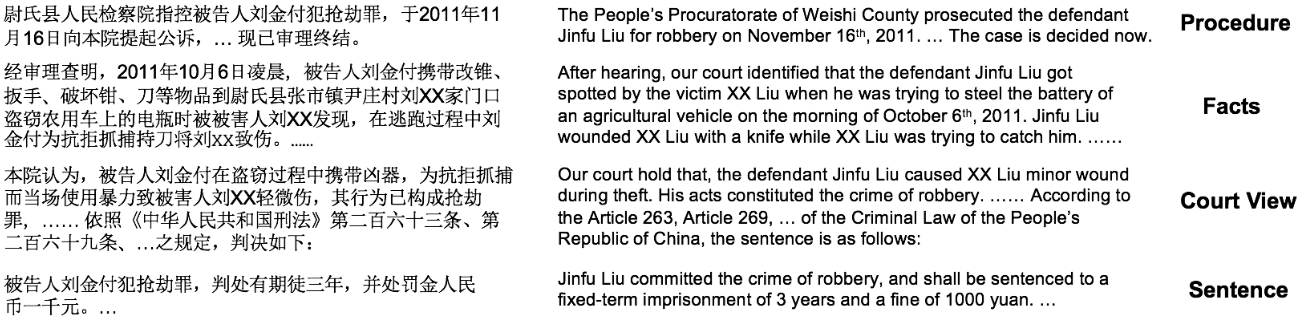
where the attention value α_t is calculated by:

$$v_t = tanh(W h_t) \tag{9}$$

$$\alpha_t = \frac{exp(v_t^T u)}{\sum_t exp(v_t^T u)} \tag{10}$$

where W is a weight matrix, and u is global attention vector that is used to distinguish import elements from unimportant ones.

[5] http://china.findlaw.cn/zuiming/



尉氏县人民检察院指控被告人刘金付犯抢劫罪，于2011年11月16日向本院提起公诉，… 现已审理终结。	The People's Procuratorate of Weishi County prosecuted the defendant Jinfu Liu for robbery on November 16th, 2011. … The case is decided now.	**Procedure**
经审理查明，2011年10月6日凌晨，被告人刘金付携带改锥、扳手、破坏钳、刀等物品到尉氏县张市镇尹庄村刘XX家门口盗窃农用车上的电瓶时被被害人刘XX发现，在逃跑过程中刘金付为抗拒抓捕持刀将刘xx致伤。......	After hearing, our court identified that the defendant Jinfu Liu got spotted by the victim XX Liu when he was trying to steel the battery of an agricultural vehicle on the morning of October 6th, 2011. Jinfu Liu wounded XX Liu with a knife while XX Liu was trying to catch him. ……	**Facts**
本院认为，被告人刘金付在盗窃过程中携带凶器，为抗拒抓捕而当场使用暴力致被害人刘XX轻微伤，其行为已构成抢劫罪，…… 依照《中华人民共和国刑法》第二百六十三条、第二百六十九条、…之规定，判决如下：	Our court hold that, the defendant Jinfu Liu caused XX Liu minor wound during theft. His acts constituted the crime of robbery. …… According to the Article 263, Article 269, … of the Criminal Law of the People's Republic of China, the sentence is as follows:	**Court View**
被告人刘金付犯抢劫罪，判处有期徒三年，并处罚金人民币一千元。…	Jinfu Liu committed the crime of robbery, and shall be sentenced to a fixed-term imprisonment of 3 years and a fine of 1000 yuan. …	**Sentence**

Figure 1: Example Judgement Document of a Chinese Criminal Case

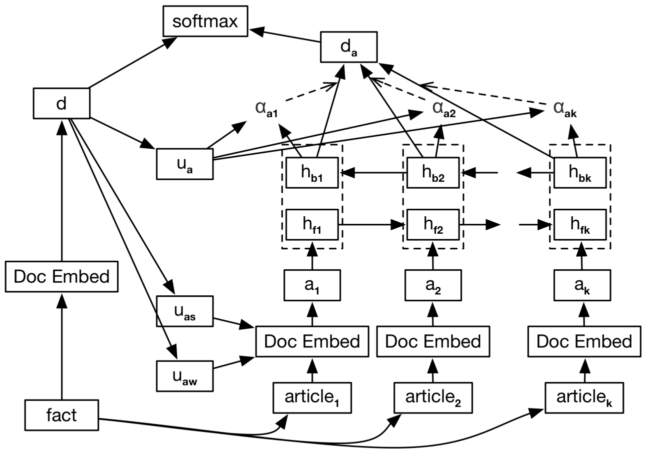


Figure 2: Model Overview

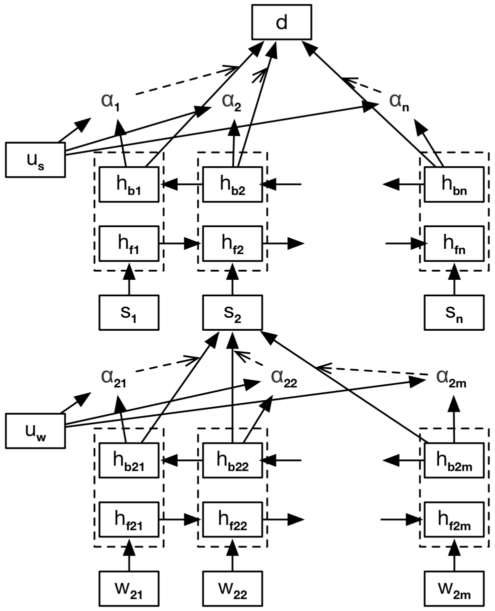


Figure 3: Document Embedding Model

Putting Everything Together As shown in Figure 3, we first use the bi-directional GRU and a word-level attention vector u_w to get the sentence embedding s_i. Then another bi-directional GRU and a sentence-level attention vector u_s is applied to the sentence sequence $[s_1, s_2, ..., s_n]$ to get the document embedding d.

4.2 Generating Output

There are two commonly used methods for multi-label classification in neural network models. The first one is to consider the multi-label classification problem with L labels as L binary classification problem. It uses sigmoid function in the output layer and binary cross entropy as loss function. This method performs well in many multi-label text classification problems [Nam *et al.*, 2014].

The second method uses the softmax function to generate outputs. It first convert the multi-label target to a probability distribution. For example, suppose there are 4 classes and one datum belongs to class 0 and class 2. This method will convert the $y = [1, 0, 1, 0]$ to $y = [0.5, 0, 0.5, 0]$, and cross entropy will be used as the loss function. After training, a threshold τ is selected and all the classes that have a score higher than τ will be considered as positive classes. This method proves to work well in the natural language query classification task [Kurata *et al.*, 2016].

In our pilot experiments, we find that the first method converges about 5 times slower than the second method in our dataset, and the second method also produces better results. We think this phenomenon happens because only a small fraction of our data are multi-label data. Therefore the second method, which uses the same output function as multi-class classification tasks, works better in our task. In this paper, we will use the second method.

Specifically, if only fact description is used, the fact embedding d is first passed to a multi-layer perceptron (MLP). The MLP output d' is further fed to a softmax function to generate the class distribution o. The cross entropy loss function is:

$$Loss_d = -\sum_{i=1}^{N}\sum_{l=1}^{L} y_{il} log(o_{il}) \tag{11}$$

where N is the number of training data, L is the number of charges, y_{il} and o_{il} are the target and predicted probability of the l^{th} charge of the i^{th} datum.

If both fact description and relevant law articles are used, the MLP input is the concatenation of of the fact embedding d and the aggregated article embedding d_a (see Section 4.4).

4.3 Extracting Relevant Articles

We consider the relevant article extraction task as a multi-label classification task. In our dataset, there are 321 law articles and we consider this task as 321 binary classification problems. In this task, we use bag-of-words TF-IDF features. First, chi-square method is applied to the original feature set for feature selection. Then SVM with linear kernel is used for the binary classification. Finally, the articles are ranked by the score output by SVM and the top k articles are kept for each judgement document.

4.4 Embed Relevant Articles

As shown in Figure 2, each article is first passed to the document embedding module to generate the article embedding $a_j, j \in [1, k]$. Different from the fact embedding module, the word level attention vector u_{aw} and sentence level attention vector u_{as} here are generated by linear transformations of the fact embedding d rather than randomly initialization:

$$u_{aw} = W_w d + b_w \tag{12}$$

$$u_{as} = W_s d + b_s \tag{13}$$

To model the phenomenon that some articles tend to coexist while others may be exclusive to each other, we also consider the k articles as a sequence, and the bi-GRU model is used to embed the context of each article. Similarly, we also generate an article level attention vector u_a by:

$$u_a = W_a d + b_a \tag{14}$$

and u_a is further used to generate article attention using Equation 9 and 10. After that, Equation 8 is used to generate the aggregated article embedding d_a.

4.5 Guided Article Attention

Note that each judgement document also contains gold standard relevant law articles that can be applied to this case. Therefore, we can use this information to guide the article attention module. Specifically, given the k articles returned by the relevant article extraction module, we want the article attention distribution $\alpha = [\alpha_1, \alpha_2, ..., \alpha_k]$ to simulate the target distribution $t \in \mathbb{R}^k$:

$$t_j = \begin{cases} 1/|\mathbb{A}|, & j \in \mathbb{A} \\ 0, & else \end{cases} \tag{15}$$

where $\mathbb{A}$ is the set of indices of the articles in the top k extracted articles that belongs to gold standard articles, and $|\mathbb{A}|$ is the size of set $\mathbb{A}$.

Therefore, the final loss function is:

$$Loss = -\sum_{i=1}^{N}(\sum_{l=1}^{L} y_{il} log(o_{il}) + \beta \sum_{j=1}^{k} t_{ij} log(\alpha_{ij})) \tag{16}$$

where the left side is the cross entropy between the target charge distribution and the predicted charge distribution defined in Equation 11, and the right side is the cross entropy between the target article distribution and the article attention distribution.

Model	Precision	Recall	F1
SVM	**93.94**/79.53	77.66/49.54	85.03/61.05
SVM_article	91.77/71.33	72.10/45.85	80.76/55.82
*SVM_gold_article**	**98.97**/94.58	95.39/83.21	97.15/88.53
*SVM_only_gold**	98.78/90.46	97.92/91.79	98.35/91.12
HAN	91.30/**83.32**	87.39/74.99	89.31/78.94
HAN_article	90.79/83.07	88.42/75.73	89.59/79.23
HAN_guide_article	91.80/82.44	**88.67/78.62**	**90.21/80.48**
HAN_gold_article*	98.78/**95.26**	**98.24/95.57**	**98.51/95.42**

Table 1: Main Results. * means we use gold standard relevant law articles, which are not available in practice.

5 Experiments

5.1 Implementation Details

We use HanLP[6] for Chinese word segmentation and POS tagging. As for word embeddings, we use the summary part of Baidu Encyclopedia, 3 million judgement documents and 3 million legal question answer pairs crawled from multiple legal forums as corpus, and the word2vec tool[7] for training. The dimension of the resultant word embedding is 100 and there are 573,353 words in total. During training and test, all the time expressions, names and charges[8] in the text are converted to 3 special tokens separately and the words not in the pre-trained word embeddings are converted to another special token. All the word embeddings remain unchanged during training except for the special tokens.

Apart form the pre-trained word embeddings, we also randomly initialize a 50 dimensional vector for each POS tag, and the POS tag embedding is concatenated with the word embedding to generate the final input. The GRU dimension is 75 and the output of the bi-directional GRU at each step is of dimension 150. The MLP between the document embedding and the softmax function has a hidden layer of size 200 and its output layer is of size 150. We keep the top 20 articles found by the relevant article extractor, and the weight of the article attention loss is 0.01. During training, the learning rate is 0.1, the batch size is 8. We also clip the gradient so that the sum of square of the L2 norms of the trainable tensors does not exceed 5. To accelerate the training, we constrain the number of sentences of each fact description to be less than 50 and the the maximum length of each fact sentence to be 50. As for articles, we constrain the maximum length of each sentence to be 30. The feature selection part in the relevant article extraction model keeps the top 2,000 features, and we use the SVM model implemented in scikit-learn [Pedregosa *et al.*, 2011] in our experiments.

5.2 Main Results

TODO: add inspiration to Chinese judiciary We compare our method with the baseline bag-of-words SVM method. The baseline method is similar to the method described in Section 4.3 except that the outputs are charges rather than articles.

[6] `https://github.com/hankcs/HanLP`

[7] `https://github.com/dav/word2vec`

[8] Although rare, sometimes the charge may appear in the fact part. This conversion ensures that we do not use this information.

	Top_5	**Top_10**	**Top_20**	**Top_30**
Recall	77.60	88.96	94.21	96.53
NDCG	80.28	84.32	86.47	87.24

Table 2: Relevant Article Extraction Result

The results is summarized in Table 1. The left side of the slash refers to the micro statistics, and the right side refers to the macro statistics. For example, the micro precision is the number of correct charge predictions divided by the total number of charge predictions. The macro precision is the average of the precision of each charge.

We can see that, the baseline SVM model proves to be a very strong baseline which achieves 85.03 micro F1. If we use both the facts and the gold standard articles of each case ($SVM\_gold\_article$), the micro F1 can be further improved to 97.15, which shows that the relevant law articles contain substantial valuable information. However, if we use the extracted top 20 articles instead of the correct articles ($SVM\_article$), the micro F1 drops significantly, which shows that the SVM cannot benefit from the additional information from relevant articles if the relevant articles contain noise. If we only use the gold standard articles of each case for classification (SVM_only_gold), the micro F1 can be further improved to 98.35, which further indicates that the SVM model has bad resistence to noise.

Since our model uses recurrent neural network for sequence encoding, it can better understand the relation between sentences and words. And due to the attention mechanism, it can further distinguish important information from unimportant ones. Therefore, when the input only contains the fact description (HAN[9]), our neural network model performs significantly better than the SVM model. Since the attention mechanism gives the model the ability to distinguish true relevant articles from wrong predictions produced by the previous step, our model can get further performance boost when the extracted top 20 articles are used ($HAN\_article$). Furthermore, if we use the gold standard articles in the training data to guide the attention procedure ($HAN\_guided\_article$), the performance can be further improved. Apart from that, if we use gold standard articles instead of extracted ones, our model also outperforms the baseline SVM model. TODO: find a name to the proposed model

5.3 Performance of Relevant Article Extraction

Our relevant article extraction module achieves 86.44% top 1 accuracy, 61.08% micro F1, and the other evaluation indicators are shown in Table 2. We can see that, the relevant article extraction task is also a hard task in itself. Although our SVM model achieves reasonable ranking performance, its prediction performance is not very good. If we use the prediction results directly in our model, we will suffer from a severe error propagation problem. Therefore, we instead let the relevant article extraction module return the top k articles, and use attention mechanism the distinguish true relevant articles from incorrect ones. Since the recall of the top 20 results has achieved 94.21%, which already includes most of the relevant articles, we use top 20 articles in our experiment.

Att_Weight	**Acc@1**	**Acc_Full**	**MAP**	**NDCG**
0	60.94	50.76	61.61	76.31
0.01	77.96	66.77	76.46	86.19
0.1	87.90	74.14	83.39	90.93
1	**92.66**	**80.43**	**88.24**	**93.81**

Table 3: Article Attention Evaluation

Model	**Precision**	**Recall**	**F1**
SVM	**95.24**/54.97	40.40/34.58	56.74/42.45
HAN	83.10/70.58	59.60/54.75	69.41/61.66
HAN_article	81.58/**72.91**	63.27/**54.95**	71.26/**62.67**
HAN_guide_article	80.25/58.51	**65.66**/47.20	**72.22**/52.25

Table 4: Results on News Data

5.4 Evaluating Article Attention

We can consider the article attention module as a re-ranking function over the extracted top k articles, and then use the gold standard articles in the top k articles to evaluate the reranking performance. Table 3 shows the model performance under different article attention loss weights (β in Equation 16, first column in Table 3). Acc_Full refers to full accuracy. For example, if there k gold standard articles in the extracted top k articles, then the full accuracy refers to the accuracy of the top k articles.

We can see that, even if there is no guide over the article attention, the model still have reasonable performance over these evaluation indicators. When attention guidance is added, the performance improves significantly, and the performance keeps increasing as β goes up. However, we find that charge classification performance does not always increases with the article attention performance, and the best result is achieved when $\beta = 0.1$. This shows that there exists a tradeoff between the benefits of more accurate article attention and the less model capacity for charge classification due to the increased emphasis on the article attention performance.

5.5 Performance on News Data

Since there are some differences between the words used in our daily life and the words used by legal practitioners, we manually annotated 100 news[10] to see how the model trained on the judgement documents performs on the fact descriptions written by normal people. The results are shown in Table 4.

We can see that although the SVM model has good performance on the judgement document data, it suffers from a significant performance drop in the news data. Recall that the SVM model is based on bag-of-words, this performance drop shows that there do exist a word usage gap between ordinary people and legal practitioners. As for our neural network models, although there also exists a performance drop, their

[9] The document embedding module is based on the HAN model.

[10] the news is randomly selected from http://www.news.cn/legal/ and http://legal.people.com.cn/

performances are much better than the SVM model. This shows that the word usage gap can be solved by word embeddings to a great extent.

Also note that the model with guided article attention has better micro statistics but worse macro statistics than the model without guided article attention. This shows that the additional article attention guidance will make the model performance better on frequent charges but worse on infrequent ones. Recall that the relevant article extraction module also uses bag-of-words features, the quality of the top k extract articles are much worse than the judgement document data (especially infrequent charges). Consider the situation where the key article related to an infrequent charge does not appear in the top k articles, but a key article related to a similar frequent charge appears. Since the article attention guidance will make the model tend to attend more on the articles related to frequent charges, the model may instead attend on the wrong article that is related to a similar frequent charge and hence performs worse on infrequent charges.

6 Conclusion

In this paper, we proposes that the public judgement documents are natural high quality training data for fact based charge prediction. Since China uses statutory law, it is reasonable to assume that finding relevant articles that can be applied to the case can help the charge prediction task. TODO: elaborate on this Based on this assumption, we propose a novel neural network model to use both the fact and the relevant articles for charge prediction. The experiments show that our model performs better than the bag-of-words baseline model as well as the neural network model that only uses fact descriptions. We also shows that the neural network model trained on the judgement document data can achieve reasonable performance on news data.

References

[Adebayo *et al.*, 2016] Kolawole John Adebayo, Guido Boella, and Luigi Di Caro. Neural reasoning for legal text understanding. *In Proc. JURIX*, 2016.

[Aletras *et al.*, 2016] Nikolaos Aletras, Dimitrios Tsarapatsanis, Daniel Preoţiuc-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.

[Bench-Capon, 1993] Trevor Bench-Capon. Neural networks and open texture. In *Proceedings of the 4th international conference on Artificial intelligence and law*, pages 292–297. ACM, 1993.

[Bruninghaus and Ashley, 2003] Stefanie Bruninghaus and Kevin D Ashley. Predicting outcomes of case based legal arguments. In *Proceedings of the 9th international conference on Artificial intelligence and law*, pages 233–242. ACM, 2003.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[Fawei *et al.*, 2016] Biralatei Fawei, Adam Wyner, and Jeff Pan. Passing a usa national bar exam: a first corpus for experimentation. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may 2016. European Language Resources Association (ELRA).

[Joachims, 1998] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning*, pages 137–142. Springer, 1998.

[Katz *et al.*, 2016] Daniel Martin Katz, II Bommarito, J Michael, and Josh Blackman. A general approach for predicting the behavior of the supreme court of the united states. *arXiv preprint arXiv:1612.03473*, 2016.

[Kim *et al.*, 2014] Mi-Young Kim, Ying Xu, and Randy Goebel. Legal question answering using ranking svm and syntactic/semantic similarity. In *JSAI International Symposium on Artificial Intelligence*, pages 244–258. Springer, 2014.

[Kim *et al.*, 2015] Mi-Young Kim, Ying Xu, and Randy Goebel. A convolutional neural network in legal question answering. *In Proc. JURISIN*, 2015.

[Kim, 2014] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[Kurata *et al.*, 2016] Gakuto Kurata, Bing Xiang, and Bowen Zhou. Improved neural network-based multi-label classification with better initialization leveraging label cooccurrence. In *Proceedings of NAACL-HLT*, pages 521–526, 2016.

[Lauderdale and Clark, 2014] Benjamin E Lauderdale and Tom S Clark. Scaling politically meaningful dimensions using texts and votes. *American Journal of Political Science*, 58(3):754–771, 2014.

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

[Martin and Quinn, 2002] Andrew D Martin and Kevin M Quinn. Dynamic ideal point estimation via markov chain monte carlo for the us supreme court, 1953–1999. *Political Analysis*, pages 134–153, 2002.

[Nam *et al.*, 2014] Jinseok Nam, Jungi Kim, Eneldo Loza Mencía, Iryna Gurevych, and Johannes Fürnkranz. Large-scale multi-label text classification—revisiting neural networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 437–452. Springer, 2014.

[Pedregosa *et al.*, 2011] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[Sim *et al.*, 2015] Yanchuan Sim, Bryan Routledge, and Noah A Smith. The utility of text: The case of amicus briefs and the supreme court. In *Proceedings of AAAI*, 2015.

[Tang *et al.*, 2015] Duyu Tang, Bing Qin, and Ting Liu. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432, 2015.

[Yang *et al.*, 2016] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016.