Learning to Predict Charges for Legal Judgment via Self-Attentive Capsule Network

Yuquan Le[1] *† , **Congqing He**[2] *† , **Meng Chen**[1] , **Youzheng Wu**[1] , **Xiaodong He**[1] and **Bowen Zhou**[1]

Abstract. With the rapid development of deep learning technology, more and more traditional industries are changed by Artificial Intelligence. The legal industry is such a popular scenario which attracts lots of researchers' interests. In this work, we focus on automatic charge prediction, which predicts the final charges according to the given fact descriptions in criminal cases. It is crucial for legal assistant systems and can help the judges improve work efficiency greatly. However, extremely imbalanced data distribution and lengthy fact descriptions make this task especially challenging. To tackle these two issues, we propose a novel model, namely Self-Attentive Capsule Network (dubbed as SAttCaps). In particular, we devise a self-attentive dynamic routing, which can not only capture long-range dependency more directly than vanilla dynamic routing, but also learn the high-level generalized features better. The experimental results on three real-world datasets demonstrate that our model significantly outperforms the baselines and creates new state-of-the-art performance. Moreover, our model performs much better than the baselines especially in the low-frequency charges and can bring 5.7% absolute improvement under F1 score.

1 INTRODUCTION

AI in law has become a popular research field and also has great value and impact in the legal industry. The task of automatic charge prediction aims to determine the final charge, such as robbery, theft or fraud, for a case by analyzing its textual fact description. It's a very important part of legal assistant system and also benefits many real-world applications. For legal professions, it can provide convenient and reliable reference and help the expert judge efficiently. For ordinary people, it can also supply legal consulting service, which is especially helpful for people unfamiliar with legal terminology and complex procedures.

Previous works usually treat automatic charge prediction as a multi-class classification problem. Most early research mainly focused on feature engineering [5, 17, 25], and made lots of efforts to extract various and efficient features from the case fact. However, this process requires numerous human work and is not easy to scale up. Owing to the success of deep learning in natural language processing tasks [8], researchers proposed to employ deep neural networks to extract legal documents features automatically [16]. Although they achieved remarkable progress in overall classification accuracy, they neglected the serious imbalanced data distribution issue in charge prediction task.

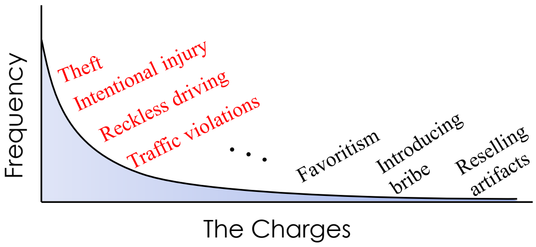


Figure 1: The statistical information on a real-world charge prediction dataset. Most of current approaches focus on high-frequency charges (shown in red text), ignoring low-frequency charges (shown in black text). This figure illustrates the distribution of criminal cases is extremely imbalanced.

In real-world scene, the distribution of criminal cases is usually extremely imbalanced. As shown in Fig 1, it's the statistical information of a real-world charge prediction dataset[3], which was published by the Chinese government from China Judgments Online[4]. There are totally 149 charges in the dataset, the most frequent 10 charges (e.g., theft, intentional injury, and traffic violations) cover 77.8% cases, while the most low-frequency 50 charges (e.g., reselling artifacts, disrupting the order of the court, and tax-escaping) only cover less than 0.5% cases and most of them only have less than 10 cases. So how to improve the performance on low-frequency (i.e. few-shot) charges becomes critically important for this task.

There are two previous representative works focusing on this issue. Hu et al. [4] proposed an attribute-attentive charge prediction model by introducing several discriminative attributes of charges, to alleviate the few-shot charges prediction problem. Ten kinds of typical discriminative attributes were artificially summarized, which can be seen as representative high-level generalized features for all charges. However, the drawback of this approach is both summarizing and annotating attributes need lots of manual work. Recently, the capsule network [23] can automatically learn a hierarchy of feature detectors via dynamic routing. A capsule is a group of neurons which uses vectors to represent an object or object part, and the orientation of the vector encodes properties of an object (like the shape/color of a face), while the length of the vector reflects its probability of existence (how likely a face with certain properties exists). These properties of capsule network could be quite appealing for catching the high-level generalized features.

[1] JD AI, Beijing, China. Emails: {leyuquan, chenmeng20, wuyouzheng1, xiaodong.he, bowen.zhou}@jd.com.
[2] JD Digits. Email: hecongqing3@jd.com.
† Equal contribution.
* Corresponding authors.

[3] Available from `https://thunlp.oss-cn-qingdao.aliyuncs.com/attribute_charge.zip`

[4] `http://wenshu.court.gov.cn`

So in another work, He et al. [2] utilized this characteristic of capsule network and proposed a Sequence Enhanced Capsule (SECaps) model. They obtained good performance in the few-shot charges without additional manual features. However, SECaps only designed an attention residual unit to capture crucial factual information. Due to independence from dynamic routing, it did not fully exploit the advantages of capsule network to learn a hierarchy of features.

Motivated by this, we propose a novel capsule-based model, namely Self-Attentive Capsule Network (dubbed as SAttCaps) for automatic charge prediction. Because the fact descriptions are generally quite lengthy, we design a novel self-attentive dynamic routing, which calculates attention weights between each pair of capsules in the low-level capsule layer, thus can capture long-range dependency more directly than vanilla dynamic routing [23]. Meanwhile, self-attentive capsule network can fully take the advantages of capsule network to learn a hierarchy of features from fact descriptions, which are similar to the representative discriminative attributes in [4]. These give our model the ability to mitigate the data imbalance issue of charge prediction task to some extent. Experimental results also prove that our approach is effective. To summarize, the main contributions of this paper are as follows:

- We propose a novel self-attentive capsule network, dubbed as SAttCaps, for charge prediction. By devising a self-attentive dynamic routing mechanism, it can better capture long-range dependency than vanilla dynamic routing and catches high-level generalized features.
- We ablate different variants of the SAttCaps model to demonstrate the rationality of self-attentive dynamic routing. We also visualize and interpret the effectiveness of the self-attentive dynamic routing with a representative case.
- We conduct extensive experiments on three real-world datasets. The experimental results demonstrate that SAttCaps model beats all baselines and creates new state-of-the-art performance. Moreover, our model outperforms previous state-of-the-art model by 5.7% absolute improvement under F1 in the low-frequency charges.

2 RELATED WORK

2.1 Capsule Network

Recently, capsule networks have been proposed by [22, 23], to improve the representation limitations of Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs). For natural language processing tasks, Yang et al. [30] explored capsule network for text classification. Wang et al. [29] proposed a capsule model based on RNN for sentiment analysis. Nguyen et al. [20] introduced a capsule network based embedding model, named CapsE, to model relationship triples. Zhang et al. [32] proposed an attention-based routing algorithm which focused on the relationship between sentence and entity. Different from above works, we apply the capsule network in charge prediction task in this paper.

2.2 Charge Prediction

Researchers have been working on automated legal judgment for a long time. Kort [10] applied quantitative methods and probability theory in analyzing judicial materials. Nagell [19] applied correlation analysis for charge prediction. Keown [6] proposed several mathematical models, including linear models, catastrophic models and the scheme of nearest neighbors, for charge prediction.

Some researchers applied machine learning based methods to legal tasks and achieved pretty good performance. Mackaay et al. [17] selected topics by clustering semantically similar n-grams as features. Liu et al. [13, 14] extracted important legal features from the documents of lawsuits and used KNN method to classify criminal charges. Lin et al. [12] manually designed key factors for case classification. Liu et al. [15] designed a text mining based method, the three-phase prediction (TPP) algorithm, for legal issues. Katz et al. [5] exploited case profiles such as terms, locations, types, and dates as features to predict the behavior of the Supreme Court of the United States. Sulea et al. [25] explored the use of lexical features and SVM ensembles to predict the law area, the ruling, and estimate the date of the ruling. However, these methods are less effective in extracting features, especially for the low-frequency charges with limited cases.

Recently, researchers begin to apply deep neural networks to legal tasks. Luo et al. [16] proposed an attention-based neural network method to jointly model the charge prediction task and the relevant article extraction task in a unified framework. Zhong et al. [33] proposed a topological multi-task learning framework which can consider the relationship between charges, fines, and the term of penalty jointly. Compared to them, our work is different in: above works focused on high-frequency charges prediction while we focused on few-shot charges.

Another line of works that discussed few-shot charge prediction are more related to our work. Hu et al. [4] proposed an attribute-attentive charge prediction model to tackle the few-shot charges. He et al. [2] proposed a sequence enhanced capsule (SECaps) model and designed an attention residual unit to capture crucial textual information. They performed well on the few-shot charges. To the best of our knowledge, SECaps model is one of the best performing models in charge prediction task. Compared with them, our work differs from: (1) we utilize self-attentive capsule network to catch high-level representative generalized features automatically, instead of designing attributes manually, (2) we catch long-range dependency more directly than dynamic routing and capture focus-related content in fact description better, which is crucial for charge prediction.

3 MODEL

The architectural overview of SAttCaps model is introduced in this section. As shown in Figure 2, SAttCaps model contains the following modules: (1) The fact encoder layer firstly maps the input fact description into the word embedding and then employs Bi-directional Long Short-Term Memory (Bi-LSTM) [1, 32] to learn fact embeddings and obtain low-level capsules. (2) These low-level capsules are fed into the self-attentive dynamic routing to produce high-level capsules. (3) The output of high-level capsules is flattened and sent to the fully connected layer to predict the charge distribution for the input case.

3.1 Problem Definition

Suppose a case contained the fact description x and the charge y, the fact description is a word sequence $x = \{x_1, x_2, \cdots, x_n\}$, where n is the sequence length, and each word $x_i \in W$, W is a fixed vocabulary. The model aims to predict the distribution of charges based on the fact description x. Meanwhile, the prediction with the highest confidence $y \in Y$ is regarded as the charge prediction result, where Y is a charge label set.

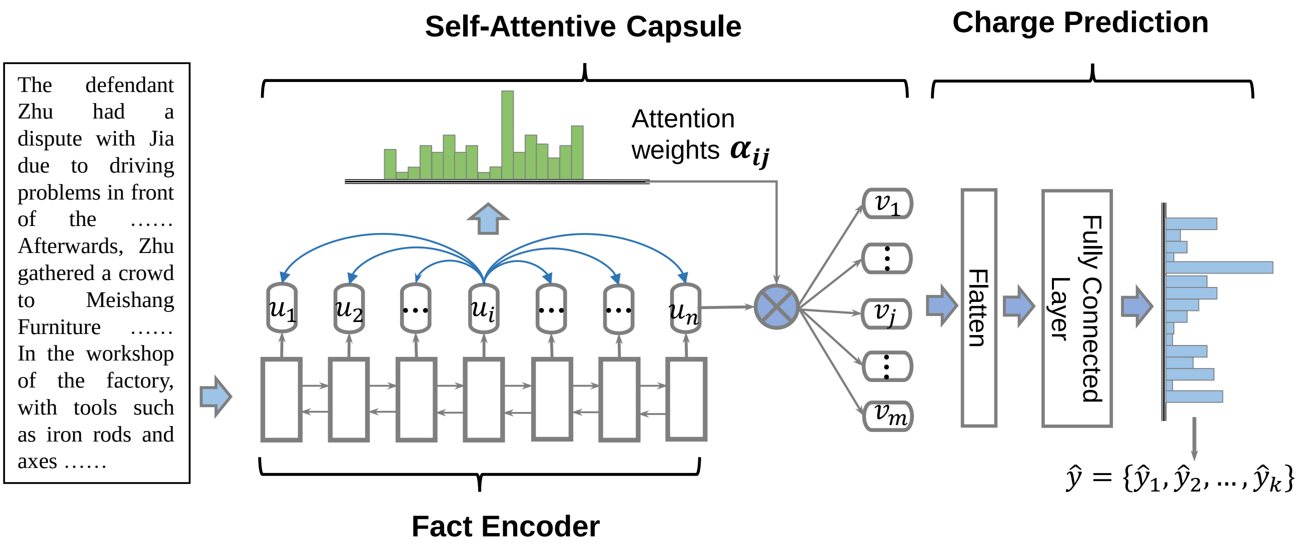

Self-Attentive Capsule
Charge Prediction
The defendant Zhu had a dispute with Jia due to driving problems in front of the …… Afterwards, Zhu gathered a crowd to Meishang Furniture …… In the workshop of the factory, with tools such as iron rods and axes ……
Attention weights $\boldsymbol{\alpha_{ij}}$
u_1
u_2
u_i
u_n
v_1
v_j
v_m
Flatten
Fully Connected Layer
$\hat{y} = \{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_k\}$
Fact Encoder

Figure 2: The architecture of SAttCaps model. $u = \{u_1, u_2, \cdots u_n\}$ is a low-level capsule set, and $v = \{v_1, v_2, \cdots, v_m\}$ represents a high-level capsule set. $\hat{y} = \{\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_k\}$ indicates the predicted charge label.

3.2 Fact Encoder Layer

The fact encoder layer first maps the fact description into the word embeddings. Afterwards, the fact encoder layer employs Bi-LSTM [1, 32] to extract contextual information, and produces low-level capsules.

Suppose the fact description of a case is a word sequence $x = \{x_1, x_2, \cdots, x_n\}$, where n is the sequence length. Each word x_i is mapped into word embedding $e_i \in \mathbb{R}^p$, where p is the dimension of word embedding. Subsequently, fact encoder layer uses Bi-LSTM including both a forward LSTM and a backward LSTM to produce low-level capsules u_i, which is similar as [31, 32]. The formula is as follows:

$$\overrightarrow{u_i} = \overrightarrow{\text{LSTM}}_i(e_1, e_2, \cdots, e_n) \quad (1)$$

$$\overleftarrow{u_i} = \overleftarrow{\text{LSTM}}_i(e_1, e_2, \cdots, e_n) \quad (2)$$

$$u_i = [\overrightarrow{u_i}, \overleftarrow{u_i}] \quad (3)$$

where $\overrightarrow{u_i}$ is the forward state and $\overleftarrow{u_i}$ is the backward state. The u_i is the concatenation of $\overrightarrow{u_i}$ and $\overleftarrow{u_i}$.

3.3 Self-Attentive Capsule Layer

Recently, the capsule network is proposed by [23] which can automatically learns a hierarchy of feature detectors via dynamic routing. A capsule is a group of neurons which uses vectors to represent an object or object part, and the orientation of vector encodes properties of an object (like the shape/color of a face), while the length of the vector reflects its probability of existence (how likely a face with certain properties exists). Group of these capsules forms a capsule layer and then these layers lead to form a capsule network. These properties of capsule network could be quite appealing for our task.

Therefore, we design a self-attentive capsule layer. In particular, we devise a self-attentive dynamic routing which is inspired by the self-attention mechanism [28]. The self-attentive dynamic routing calculates attention weights between each pair of capsules in a low-level capsule layer, thus can capture long-range dependency more directly than dynamic routing [23]. Then the low-level capsules are fed into self-attentive dynamic routing to generate high-level capsules. By producing representative high-level capsules, it can catch the high-level generalized features and captures focus-related content in legal text better. Thus, it can better handle the legal texts with lengthy fact descriptions in our task.

The self-attentive dynamic routing is summarized in Algorithm 1. Formally, the low-capsules $u = \{u_1, \cdots, u_n\}$ of n capsules are obtained by fact encoder layer, where $u_i \in \mathbb{R}^d$ represents low-capsule i. The coupling coefficients w between i-th low-level capsule and all the high-level capsules $v = \{v_1, \cdots v_m\}$ sum to 1. They are determined by a "softmax" function whose initial logits are b_{ij}, the log prior probabilities that capsule u_i should be coupled to capsule v_j. Besides, we design attentive weights α to represent the importance distribution of the low-level capsules. The high-level capsule v_j is computed as the weighted sum of a linearly transformed all the low-level capsules u by multiplying the coupling coefficients w.

$$v_j = \textbf{squash}\Big(\sum_i w_{ij} \cdot (\sum_{j'} \alpha_{ij'} \cdot u_{j'} W^V_{j'})\Big) \quad (4)$$

where α_{ij} is an attentive weight which is computed by Eq. (6). u_j is j-th low-level capsule, and $W^V \in \mathbb{R}^{d \times d_z}$ is a linear transformation matrix. w_{ij} is coupling coefficients, $v_j \in \mathbb{R}^{d_z}$ is j-th high-level capsule. Each high-level capsule is applied with a non-linear "squash" function.

$$\textbf{squash}(v_j^{'}) = \left(\frac{\|v_j^{'}\|^2}{1 + \|v_j^{'}\|^2} \cdot \frac{v_j^{'}}{\|v_j^{'}\|}\right) \quad (5)$$

Here, each attention weights α_{ij} is computed as:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{n} \exp(e_{ik})} \quad (6)$$

and e_{ij} is computed as multiplying a linearly transformed low-level capsule u_i by a linearly transformed low-level capsule u_j.

$$e_{ij} = \frac{(u_i W_i^Q)(u_j W_j^K)^T}{\sqrt{d_z}} \quad (7)$$

where $W^Q \in \mathbb{R}^{d \times d_z}$ and $W^K \in \mathbb{R}^{d \times d_z}$ are linear transformation matrices of low-level capsules. d_z is a scaling factor [28]. Finally, the high-level capsules v are computed with the Algorithm 1.

```
Algorithm 1: Self-Attentive Dynamic Routing
INPUT    : low-level capsules: $u$;
           iterative number: $r$
OUTPUT: high-level capsule: $v$
Initialize the logits of coupling coefficients $b_{ij} = 0$.
for $r$ iterations do
    $w_i = \text{softmax}(b_i)$,
    $e_{ij} = \frac{(u_i W_i^Q)(u_j W_j^K)^T}{\sqrt{d_z}}$,
    $\alpha_{ij} = \text{softmax}(e_{ij})$,
    $v_j = \textbf{squash}\left(\sum_i w_{ij} \cdot (\sum_{j'} \alpha_{ij'} \cdot u_{j'} W_{j'}^V)\right)$,
    $b_{ij} = b_{ij} + (\sum_{j'} \alpha_{ij'} u_{j'} W_{j'}^V) \cdot v_j$.
return $v_j$;
```

3.4 Charge Predicting Layer

The output of the self-attentive capsule layer are high-level capsules which extract different semantic features from the fact descriptions. The high-level capsules are flattened into a list of capsules and fed into a fully connected layer to predict the probability of charges. Hereafter, the output of representations is passed to a softmax classifier, to obtain the probability of charge predictions. In practice, charge prediction is confronted with extremely imbalanced data distributions. Lin et al. [11] proposed a novel Focal Loss (FL) which can alleviate the imbalance data distribution problem. Inspired by this, we extend the FL to the multi-class scenario and transfer the FL to the few-shot charge prediction task. Therefore, we use the FL as loss function in our model. The formula is as follows:

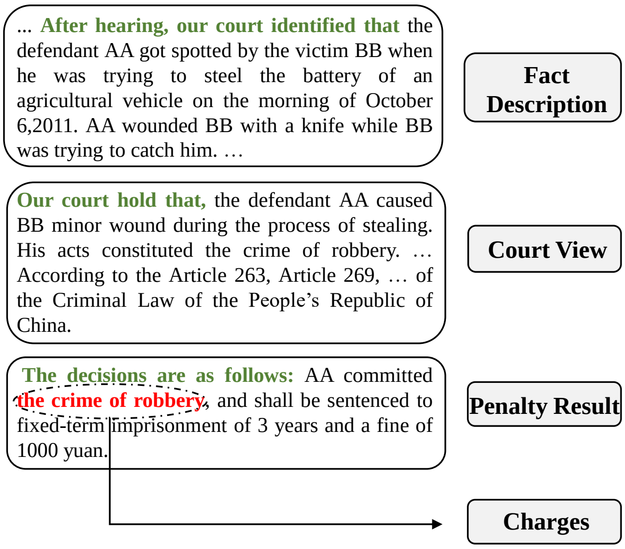
$$L_{FL} = -\sum_{t \in T} \sum_{c=1}^{C} \mu (1 - \hat{y}_t)^{\gamma} \log(\hat{y}_t) \tag{8}$$

where $\hat{y}_t \in [0, 1]$ is the model's estimated probability which is computed by a softmax function. T is the training data, C is the number of classes, μ is μ-balanced variant of the FL. $(1 - \hat{y}_t)^{\gamma}$ is a modulating factor and γ ($\gamma \geq 0$) is a tunable focusing parameter which increases the effect of the modulating factor. In our experiments, the hyper-parameters of the FL are kept consistent with [11].

4 EXPERIMENTS

4.1 Datasets and Metrics

Datasets: Following [4], we conduct experiments on three real-world datasets published by the Chinese government from China Judgments Online, to demonstrate the effectiveness of SAttCaps model on criminal charge prediction. As illustrated in Fig. 3, each case includes several parts: fact description, court view, and penalty result. We select the fact description of each case as input and use the charge from the penalty result as output. The three datasets are Criminal-S (small), Criminal-M (medium) and Criminal-L (large) respectively, which contain the same amount of charges but different amount of cases. The detailed statistics are presented in Table 1. Please notice that the fact description is usually very lengthy. According to our statistics on Criminal-L dataset, more than 50% of fact descriptions consist of more than 250 words.


... After hearing, our court identified that the defendant AA got spotted by the victim BB when he was trying to steel the battery of an agricultural vehicle on the morning of October 6,2011. AA wounded BB with a knife while BB was trying to catch him. ...

Our court hold that, the defendant AA caused BB minor wound during the process of stealing. 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 decisions are as follows: AA committed the crime of robbery, and shall be sentenced to fixed-term imprisonment of 3 years and a fine of 1000 yuan.

Figure 3: A legal judgement document example for a criminal case in our dataset [16]. Names are anonymized as AA and BB. The green texts refer to the clauses that usually indicate the beginning of the fact part, the court view part and the penalty result, respectively. Charges are extracted with regular expressions from the penalty results.

Table 1: The statistics of three datasets.

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

Evaluation Metrics: Following [4, 16], the performance of charge prediction task is measured by the classification accuracy (Acc), macro-precision (MP), macro-recall (MR), and macro-F1 (F1). Please notice that Acc can only reflect the overall performance of the model, which might be dominated by high-frequency charges. However, the MP, MR, F1 are more fair for evaluating model on the imbalanced datasets, especially for low-frequency classes.

4.2 Baselines

Following [4], we compare SAttCaps model with several competitive text classification models and previous state-of-the-art charge prediction models:

TFIDF-SVM: The TFIDF-SVM uses term-frequency inverse document frequency (TFIDF) [24] to represent the fact description as input and employs SVM [27] as classifier.

CNN: The convolutional neural network (CNN) [8] uses convolution with multiple filter widths and pooling operations.

LSTM: The long short-term memory networks (LSTM) [3] uses a two-layer LSTM with a max-pooling operation to encode the fact descriptions and uses two fully connected layers with a softmax as the classifier.

Table 2: Charge prediction results on three datasets, where the best results are highlighted in bold.

Models	Criminal-S				Criminal-M				Criminal-L			
	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
Fact-Law Attention	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	93.4	66.7	69.2	64.9	94.4	68.3	69.2	67.1	95.8	75.8	73.7	73.1
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**

Table 3: F1 of different models on Criminal-S with different frequencies, where the best results are highlighted in bold.

Charge Type	Low-Frequency	Medium-Frequency	High-Frequency
Charge Number	49	51	49
Attribute-Attentive	49.7	60.0	85.2
SECaps	53.8	65.5	89.0
SAttCaps	**59.5**	**67.8**	**89.4**

CNN-Capsule: The CNN-Capsule uses convolution with filter widths $(2, 3, 4, 5)$ to produce low-level capsules, and then feeds into dynamic routing [30].

Fact-Law Attention Model: Luo et al. [16] proposed an attention-based neural network method for charge prediction task by combining fact descriptions and extracting relevant law articles.

Attribute-attentive Model: Hu et al. [4] proposed an attribute-attentive charge prediction model by introducing several discriminative attributes of charges for the few-shot charges prediction.

SECaps Model: He et al. [2] proposed a sequence enhanced capsule model to predict few-shot charges with limited cases.

4.3 Implementation Details

Following the experimental setup of [4], all the fact descriptions of cases are processed by THULAC[5] [26] for word segmentation and the maximum length of all sequences is set to 500. The word embeddings are pre-trained by word2vec [18] with the dimension of 100. For a fair comparison with [2, 4], the hidden size of fact encoder layer is set to 200 for each direction in Bi-LSTM. In the self-attentive capsule layer, the number and dimension of the high-level capsule are set to 8 and 16 respectively, the number of iterations r is set to 3, which is the same as [23].

Inspired by [7], Adam [9] outperforms Stochastic Gradient Descent (SGD) [21] in both training and generalization metrics in the initial portion of training, but then the performance stagnates. In order to improve the generalization performance of SAttCaps model, we introduce a hybrid optimizer [7] to minimize the Focal Loss. We begin training with Adam then switch to SGD when appropriate, which is similar to Switches from Adam to SGD method (SWATS) [7]. Specifically, SAttCaps model is trained with Adam in the first 25 epochs and switch to SGD in the last 5 epochs. Furthermore, the model monitors its performance on the Dev dataset and keeps its best accuracy score on the Dev dataset for each epoch. Once training is finished, we employ the model with the best accuracy on the Dev set as our final model and evaluate the performance on the Test set.

In addition, the experimental settings of baselines are as follows. The hyper-parameter settings of the TFIDF-SVM, CNN, and LSTM model remain unchanged with [4]. For the existing state-of-the-art models, the parameters of models are consistent with the original papers and the results are collected from [2, 4, 16]. Considering deep neural networks training is a stochastic process, we ran multiple trials for each experiment and an average of multiple trials was reported to avoid bias introduced by randomness.

5 EXPERIMENTAL RESULTS

5.1 Performance Comparison

Table 2 shows the experimental results on three datasets, where the best result for each metric is highlighted in bold. These models employ Adam [9] to minimize the loss function, except TFIDF-SVM and SAttCaps model. From the table we can derive the following interesting conclusions: (1) Our proposed model SAttCaps outperforms all the baselines. Compared to state-of-the-art [2], we still got 2.8%, 6.8%, and 3.0% absolute improvements on F1 scores for all three test sets correspondingly. This proves that our model has a very strong ability to catch the important semantic information in lengthy fact descriptions. Please notice that the Acc improvement is not so huge, which is caused by the imbalanced data distribution in the test sets, just as mentioned in Section 4.1. (2) Approaches (SAttCaps, SECaps, and Attribute-Attentive) that are designed for dealing with few-shot charges perform much better than those not. It proves that tackling the data imbalance issue is crucial for solving the charge prediction task. (3) Please also notice that performance of CNN-Capsule model is comparable with Attribute-Attentive on Criminal-M and Criminal-L, which indicates capsule network has the advantage in catching the high-level generalized features automatically.

[5] https://github.com/thunlp/THULAC-Python

Table 4: Performance of different ablated models on three datasets, where the best results are highlighted in bold.

Models	Criminal-S				Criminal-M				Criminal-L			
	Acc	MP	MR	F1	Acc	MP	MR	F1	Acc	MP	MR	F1
Capsule (LSTM-based)	94.8	68.7	70.0	68.3	95.5	70.2	68.1	68.6	96.1	79.4	75.7	76.0
SAttCaps (LSTM-based)	94.9	71.0	70.5	69.8	95.8	74.2	72.0	72.3	96.2	84.1	81.0	81.1
Capsule (Bi-LSTM-based)	94.9	70.4	71.3	69.6	95.6	73.6	71.7	71.8	96.0	81.2	75.7	76.7
SAttCaps$_{\text{CE+Adam}}$	95.1	72.4	72.5	71.3	96.0	74.8	72.6	72.4	96.2	82.8	79.9	80.1
SAttCaps$_{\text{FL+Adam}}$	95.1	72.6	**72.7**	71.5	96.0	76.9	74.6	74.6	96.4	84.0	80.4	81.1
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**

5.2 Few-shot Performance Comparison

In this subsection, we compare SAttCaps model with two competitive models [2, 4], to further demonstrate the performance of SAttCaps model on dealing with imbalanced charge prediction, especially for few-shot charges. Meanwhile, the same as [2, 4], we use F1 to evaluate the model performance. F1 can reflect the performance of the model on the imbalanced problem, especially on low-frequency charges. Following [2, 4], the Criminal-S is divided into three parts according to the frequency of charges. Specifically, the charges with ≤ 10 cases are low-frequency, and there are totally 489 cases. The charges with > 100 cases are high-frequency, and there are 74520 cases. The others belong to medium-frequency, and these charges include 2034 cases.

The experimental results on Criminal-S are shown in Table 3. It can be seen from this table that the proposed method in this paper performs better than Attribute-Attentive model [4] and SECaps model [2] on all frequency subsets. Especially on low-frequency, SAttCaps model outperforms SECaps model by 5.7% absolute improvement in terms of F1. These evidences demonstrate that SAttCaps model is more effective and competitive for alleviating the imbalanced charge prediction problem.

5.3 Ablation Study

In this subsection, we ablate different variants of SAttCaps model on three datasets, to evaluate the effectiveness of our SAttCaps model. We compare SAttCaps model with the following variant models using the same experimental setup.

Capsule (LSTM-based): it use LSTM with 200 hidden nodes as fact encoder and then feeds into capsule network [30].

SAttCaps (LSTM-based): it is the same as SAttCaps model but using LSTM instead of Bi-LSTM to encode the fact descriptions.

Capsule (Bi-LSTM-based): it applies Bi-LSTM with 200 hidden nodes as fact encoder, and then feeds into capsule network [30].

Moreover, to show the effectiveness of Focal Loss [11] and SWATS optimizer [7] of SAttCaps model, we conduct another group of ablation study.

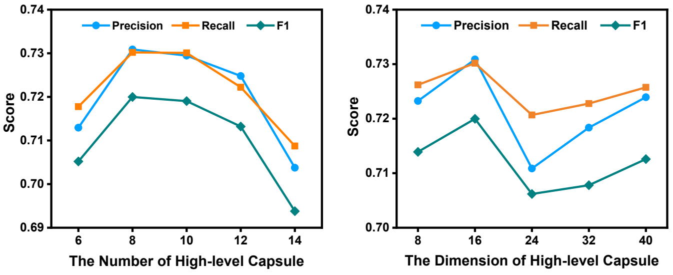
SAttCaps$_{\text{FL+Adam}}$: it employs Adam as optimizer function to minimize the loss function, and other settings remain unchanged.

SAttCaps$_{\text{CE+Adam}}$: it uses cross-entropy (CE) as loss function and Adam as optimizer, while other settings remain unchanged.

Table 4 presents the experimental results. We have the following observations: (1) Capsule networks with (Bi-)LSTMs perform better than that with CNNs, which indicates LSTMs are better than CNN in capturing long-range dependency. (2) SAttCaps (LSTM-based) is better than Capsule (LSTM-based), and SAttCaps (Bi-LSTM-based) is better than Capsule (Bi-LSTM-based), which proves the effectiveness of self-attentive dynamic rooting mechanism. (3) SAttCaps (LSTM-based) is better than Capsule (Bi-LSTM-based), which further demonstrates the self-attentive dynamic routing, which calculates attention weights between each pair of capsules in the low-level capsule layer, thus can capture long-range dependency more directly than dynamic routing.

For a fair comparison with [2, 4], we compare SAttCaps model with different loss function and optimizer on three datasets. It can be seen that, (4) SAttCaps$_{\text{FL+Adam}}$ model outperforms most of the existing baselines in Section 5.1, which indicates that the SAttCaps model without SWATS optimizer can still perform well. Furthermore, it is also observed that SAttCaps with SWATS optimizer achieves better performance. This indicates SWATS optimizer can improve the generalization performance of the model. (5) After removing FL, SAttCaps$_{\text{CE+Adam}}$ model still perform better than all baselines in Section 5.1 on three datasets. What's more, SAttCaps$_{\text{FL+Adam}}$ model outperforms SAttCaps$_{\text{CE+Adam}}$ model on all datasets. This proves the effectiveness of Focal Loss in alleviating the imbalanced problem.

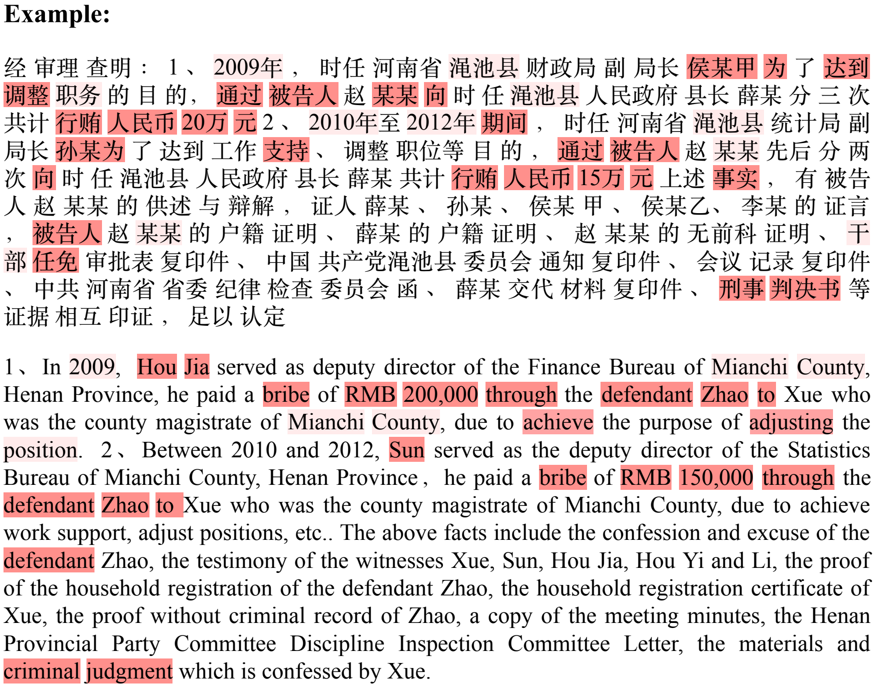
5.4 Impact of Hyper-parameters



(a) (b)

Figure 4: (a): Performance of SAttCaps with different number of high-level capsule (fixing the dimension of high-level capsule to 16). (b): Performance of SAttCaps with different dimension of high-level capsule (fixing the number of high-level capsule to 8).

Our model has two important hyper-parameters: the number of high-level capsule and the dimension of the high-level capsule. Here we study the impact of these two hyper-parameters on the performance of our model. Figure 4 shows the detailed results, which are experimented on the Criminal-S dataset. Figure 4(a) shows that, our model gains the best performance (MP/MR/F1) when the number of high-level capsule is 8. Figure 4(b) shows that when the dimension of high-level capsule is set to 16, the performance of model reaches the best. It seems that when the dimension exceeds



Example:

经 审理 查明： 1、 2009年， 时任 河南省 渑池县 财政局 副 局长 侯某甲 为 了 达到 调整 职务 的 目 的， 通过 被告人 赵 某某 向 时 任 渑池县 人民政府 县长 薛某 分 三 次 共计 行贿 人民币 20万 元 2、 2010年至 2012年 期间， 时任 河南省 渑池县 统计局 副 局长 孙某为 了 达到 工作 支持、 调整 职位等 目 的， 通过 被告人 赵 某某 先后 分 两 次 向 时 任 渑池县 人民政府 县长 薛某 共计 行贿 人民币 15万 元 上述 事实， 有 被告 人 赵 某某 的 供述 与 辩解， 证人 薛某、 孙某、 侯某 甲、 侯某乙、 李某 的 证言， 被告人 赵 某某 的 户籍 证明、 薛某 的 户籍 证明、 赵 某某 的 无前科 证明、 干 部 任免 审批表 复印件、 中国 共产党渑池县 委员会 通知 复印件、 会议 记录 复印件、 中共 河南省 省委 纪律 检查 委员会 函、 薛某 交代 材料 复印件、 刑事 判决书 等 证据 相互 印证， 足以 认定

1、In 2009, Hou Jia served as deputy director of the Finance Bureau of Mianchi County, Henan Province, he paid a bribe of RMB 200,000 through the defendant Zhao to Xue who was the county magistrate of Mianchi County, due to achieve the purpose of adjusting the position. 2、Between 2010 and 2012, Sun served as the deputy director of the Statistics Bureau of Mianchi County, Henan Province , he paid a bribe of RMB 150,000 through the defendant Zhao to Xue who was the county magistrate of Mianchi County, due to achieve work support, adjust positions, etc.. The above facts include the confession and excuse of the defendant Zhao, the testimony of the witnesses Xue, Sun, Hou Jia, Hou Yi and Li, the proof of the household registration of the defendant Zhao, the household registration certificate of Xue, the proof without criminal record of Zhao, a copy of the meeting minutes, the Henan Provincial Party Committee Discipline Inspection Committee Letter, the materials and criminal judgment which is confessed by Xue.

Figure 5: Visualization of self-attentive dynamic routing by a *"introducing bribery"* case. The deeper color of words denotes that more information is routed to the the corresponding high-level capsule.

40, the performance starts to increase. We argue it's possible that the larger the dimension is, the more capabilities the model has. Correspondingly, the computational complexity becomes higher. Therefore, we set the number of high-level capsule to 8 and the dimension of high-level capsule to 16 in our experiments to balance the performance and training cost. These results further demonstrate the rationality of the hyper-parameter settings in Section 4.3.

5.5 Case Study

In order to show the proposed self-attentive dynamic routing can better capture focus-related content of fact description, we visualized a representative case to give an intuitive illustration of how much information of the low-level capsules sends to high-level capsules. We choose the charge, "introducing bribery", which only appears 10 times in Criminal-S. In fact, it is hard to determine whether a case is "*bribery*" or "*introducing bribery*" since they are very similar. "Bribery" is the act of giving money, goods, or other forms of compensation to a recipient in exchange for some kind of influence or action in return. "Introducing bribery" is the act as a go-between (matchmaker) for the briber on purpose to facilitate the bribery transaction, and finally making bribery realize.

As shown in Figure 5, the *defendant Zhao* is convicted of *introducing bribery* in the case. We observe that SAttCaps model pays more attention to words which are related to the charge, such as "*defendant*", "*bribery*". Apparently, the self-attentive dynamic routing also has assigned heavier weights to "*through*" and "*to*", which is crucial for judging the final charge in this case. Therefore, SAttCaps model predicts this case as "*introducing bribery*" correctly according to these key information. In conclusion, the visualization of self-attentive dynamic routing shows SAttCaps model can effectively capture focus-related content of fact description, which is crucial for charge prediction.

6 CONCLUSION

In this paper, we focus on automatic charge prediction, which predicts the final charges according to the given fact descriptions in criminal cases. In order to alleviate the problem of extremely imbalanced data distribution and quite long fact descriptions, we propose a novel model, namely Self-Attentive Capsule Network (dubbed as SAttCaps). In particular, we devise a self-attentive dynamic routing, which can not only capture long-range dependency more directly than dynamic routing, but can also learn the high-level generalized features better. The experimental results on three real-world datasets demonstrate that our model is significantly better than baselines and create new state-of-the-art performance. Moreover, our model performs much better in the low-frequency (i.e. few-shot) charges. In the future, we will focus on more complicated criminal cases, such as cases with multiple defendants and charges, which are quite challenging at the current stage.

ACKNOWLEDGEMENTS

This work is partially supported by Beijing Academy of Artificial Intelligence (BAAI).

REFERENCES

[1] Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann, 'Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm', in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1615–1625, Copenhagen, Denmark, (September 2017). Association for Computational Linguistics.

[2] Congqing He, Li Peng, Yuquan Le, Jiawei He, and Xiangyu Zhu, 'Secaps: A sequence enhanced capsule model for charge prediction', in *Artificial Neural Networks and Machine Learning – ICANN 2019: Text and Time Series*, eds., Igor V. Tetko, Věra Kůrková, Pavel Karpov,

and Fabian Theis, pp. 227–239, Cham, (2019). Springer International Publishing.

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

[4] Zikun Hu, Xiang Li, Cunchao Tu, Zhiyuan Liu, and Maosong Sun, 'Few-shot charge prediction with discriminative legal attributes', in *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 487–498, Santa Fe, New Mexico, USA, (August 2018). Association for Computational Linguistics.

[5] Daniel Martin Katz, Michael J Bommarito II, and Josh Blackman, 'A general approach for predicting the behavior of the supreme court of the united states', *PloS one*, **12**(4), e0174698, (2017).

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

[7] Nitish Shirish Keskar and Richard Socher, 'Improving generalization performance by switching from adam to sgd', *arXiv preprint arXiv:1712.07628*, (2017).

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

[9] Diederik P Kingma and Jimmy Ba, 'Adam: A method for stochastic optimization', *arXiv preprint arXiv:1412.6980*, (2014).

[10] 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).

[11] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár, 'Focal loss for dense object detection', in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, (2017).

[12] 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', in *International Journal of Computational Linguistics & Chinese Language Processing*, volume 17, pp. 49–68, (2012).

[13] Chao-Lin Liu, Cheng-Tsung Chang, and Jim-How Ho, 'Case instance generation and refinement for case-based criminal summary judgments in chinese', (2004).

[14] Chao-Lin Liu and Chwen-Dar Hsieh, 'Exploring phrase-based classification of judicial documents for criminal charges in chinese', in *International Symposium on Methodologies for Intelligent Systems*, pp. 681–690. Springer, (2006).

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

[16] Bingfeng Luo, Yansong Feng, Jianbo Xu, Xiang Zhang, and Dongyan Zhao, 'Learning to predict charges for criminal cases with legal basis', in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2727–2736, Copenhagen, Denmark, (September 2017). Association for Computational Linguistics.

[17] Ejan Mackaay and Pierre Robillard, *Predicting judicial decisions: The nearest neighbour rule and visual representation of case patterns*, 1974.

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

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

[20] Dai Quoc Nguyen, Thanh Vu, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung, 'A Capsule Network-based Embedding Model for Knowledge Graph Completion and Search Personalization', in *Proceedings of the 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, (2019).

[21] Herbert Robbins and Sutton Monro, 'A stochastic approximation method', *The annals of mathematical statistics*, 400–407, (1951).

[22] Sara Sabour, Nicholas Frosst, and G Hinton, 'Matrix capsules with em routing', in *6th International Conference on Learning Representations, ICLR*, (2018).

[23] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton, 'Dynamic routing between capsules', in *Advances in Neural Information Processing Systems 30*, eds., I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, 3856–3866, Curran Associates, Inc., (2017).

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

[25] Octavia-Maria Sulea, Marcos Zampieri, Shervin Malmasi, Mihaela Vela, Liviu P Dinu, and Josef van Genabith, 'Exploring the use of text classification in the legal domain', *arXiv preprint arXiv:1710.09306*, (2017).

[26] Maosong Sun, Xinxiong Chen, Kaixu Zhang, Zhipeng Guo, and Zhiyuan Liu, 'Thulac: An efficient lexical analyzer for chinese', Technical report, Technical Report, (2016).

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

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, 'Attention is all you need', in *Advances in Neural Information Processing Systems*, pp. 5998–6008, (2017).

[29] Yequan Wang, Aixin Sun, Jialong Han, Ying Liu, and Xiaoyan Zhu, 'Sentiment analysis by capsules', in *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pp. 1165–1174. International World Wide Web Conferences Steering Committee, (2018).

[30] Min Yang, Wei Zhao, Jianbo Ye, Zeyang Lei, Zhou Zhao, and Soufei Zhang, 'Investigating capsule networks with dynamic routing for text classification', in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3110–3119, Brussels, Belgium, (2018). Association for Computational Linguistics.

[31] Ningyu Zhang, Shumin Deng, Zhanling Sun, Xi Chen, Wei Zhang, and Huajun Chen, 'Attention-based capsule networks with dynamic routing for relation extraction', in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 986–992, Brussels, Belgium, (October-November 2018). Association for Computational Linguistics.

[32] Xinsong Zhang, Pengshuai Li, Weijia Jia, and Hai Zhao, 'Multi-labeled relation extraction with attentive capsule network', *Proceedings of the AAAI Conference on Artificial Intelligence*, **33**, 7484–7491, (07 2019).

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