applied sciences

MDPI

Article

Few-Shot Charge Prediction with Data Augmentation and Feature Augmentation

Peipeng Wang, Xiuguo Zhang * and Zhiying Cao *

School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China; wpp_678@dlmu.edu.cn
* Correspondence: zhangxg@dlmu.edu.cn (X.Z.); czysophy@dlmu.edu.cn (Z.C.)

Featured Application: This study aims to effectively analyze legal cases, and the findings serve as a reference for the business domain.

Abstract: The task of charge prediction is to predict the charge based on the fact description. Existing methods have a good effect on the prediction of high-frequency charges, but the prediction of low-frequency charges is still a challenge. Moreover, there exist some confusing charges that have relatively similar fact descriptions, which can be easily misjudged. Therefore, we propose a model with data augmentation and feature augmentation for few-shot charge prediction. Specifically, the model takes the text description as the input and uses the Mixup method to generate virtual samples for data augmentation. Then, the charge information heterogeneous graph is introduced, and a novel graph convolutional network is designed to extract distinguishability features for feature augmentation. A feature fusion network is used to effectively integrate the charge graph knowledge into the fact to learn semantic-enhanced fact representation. Finally, the semantic-enhanced fact representation is used to predict the charge. In addition, based on the distribution of each charge, a category prior loss function is designed to increase the contribution of low-frequency charges to the model optimization. The experimental results on real-work datasets prove the effectiveness and robustness of the proposed model.

Keywords: charge prediction; Mixup; graph convolutional network; loss function

Citation: Wang, P.; Zhang, X.; Cao, Z. Few-Shot Charge Prediction with Data Augmentation and Feature Augmentation. *Appl. Sci.* **2021**, *11*, 10811. https://doi.org/10.3390/app112210811

Academic Editor: Rui Araújo

Received: 22 September 2021
Accepted: 13 November 2021
Published: 16 November 2021

1. Introduction

Charge prediction refers to determining the final charge of the case (such as "theft" and "robbery") according to the text description of the case. It plays an irreplaceable role in the judicial system. It not only provides convenient reference for legal experts, but also provides important legal advice for ordinary people who are not familiar with legal knowledge [1].

Automatic charge prediction has been studied for decades. Early researchers used statistical methods to analyze the factual factors affecting decision making and predict charges [2]. It was only effective for small-scale data and cases with obvious characteristics, which was difficult to extend to general cases. Later, with the development of machine learning algorithms, researchers [3–5] further extracted effective features from fact description and made predictions by machine learning methods, such as Naive Bayesian Model, SVMs, or K-NN. However, these methods rely heavily on manual features, which are difficult to gather on a bigger dataset. In recent years, with the successful usage of deep learning methods on speech, computer vision and natural language processing, the neural network is used to model legal documents [1].

However, charge prediction is not a simple task. There are still two main challenges in the real scene: few-shot charges and confusing charges. Additionally, some few-shot charges often have the characteristics of confusing, which undoubtedly adds new difficulties to the research. On the one hand, in the real datasets, the number of cases of

various charges is extremely unbalanced, and the ten most common charges cover 78% of cases; the uncommon 50 charges cover less than 0.5% of the cases. In case of insufficient training data, the general deep learning model performs poorly. To solve this problem, Zhang et al. [1] used different convolution neural networks to extract fine-grained features and used capsule network for feature fusion. He et al. [6] effectively used focal loss function to predict few-shot charges. However, there are still some limitations in this study: (1) There is still a lack of sufficient data to train the neural network model, and the fine-grained features extracted by the model cannot be used to distinguish some confusing charges [7]. (2) Many studies do not consider the effect of the prior distribution of the charges on the cross-entropy loss function, so it is difficult to classify few-shot charges effectively.

On the other hand, there are many confusing charges in the real dataset, such as robbery and snatch. The cases of these charges are similar, but the judgment results are different. How to capture the subtle differences in the cases is a challenging task. Xu et al. [8] proposed a novel graph neural network to automatically extract distinguishable features between legal articles. Some researchers considered introducing external knowledge, for example, Hu et al. [9] introduced 10 discriminative attributes of charges and used the attention mechanism to obtain fact descriptions related to specific attributes, which offered signals for distinguishing confusing charges. However, summaries and notes require a lot of manual labeling, which relies too much on expert knowledge to extract sufficient distinguishable features when training data is small. In practice, we hope that even in the case of data imbalance, the model can still effectively extract distinguishable features and accurately predict the charges.

Therefore, we propose a novel model with data augmentation and feature augmentation for few-shot charge prediction. From the perspective of data augmentation, different from the existing fine-grained feature extraction methods, this paper draws on the Mixup data augmentation method in image classification [10]. We propose a text data augmentation strategy with Mixup, which enlarges the training samples by linear interpolation from word level and sentence level in the hidden layer space of text representation, so as to enrich the feature diversity and improve the training effect of the model on low-frequency charges.

Then, from the perspective of feature augmentation: Firstly, the criminal law divides the charges into many categories, such as “crimes against national security” and “crimes against public security”. Each category is subdivided into specific charges, and the specific charges have their own definitions. Therefore, the category charge definition can be formalized into a hierarchical structure, as shown in Figure 1. By exploring the similar relationship between different charges under the same category and the co-occurrence relationship between charge definition words, this paper constructs multiple groups of charge label heterogeneous graphs. Then, in order to capture the high-order features between heterogeneous graphs, a Dropedge graph convolutional network (DGCN) is designed to capture the distinguishable features between charges without over-smoothing and generate charge graph information. The confusing charges can be effectively distinguished by analyzing this information. At the same time, because each heterogeneous graph contains multiple charges, based on the information interaction between charges, knowledge can be transferred from high-frequency charges to low-frequency charges, so the prediction effect of low-frequency charges can be further improved. A feature fusion network is used to effectively integrate the charge graph information into the case to learn semantic-enhanced fact representation. Finally, this paper uses the semantic-enhanced fact representation to predict final charges. In addition, based on the prior distribution of different charges, this paper constructs a category prior loss function to pay more attention to low-frequency charges in the training process. To demonstrate the advantages of this paper’s model, we conduct several experiments on real-work datasets. The experimental results show that the model in this paper outperforms other baseline models and achieves significant improvements for few-shot charges and confusing charges.

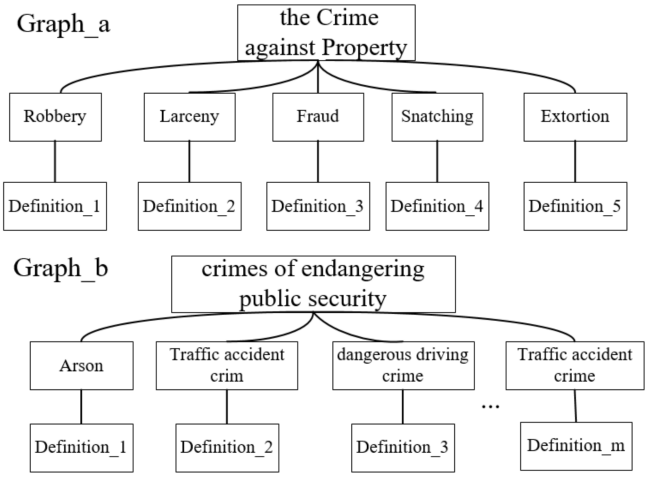

Figure 1. Charge hierarchy label structure.

The main contributions of this paper are as follows:

(1) We adopt the Mixup method to augment the training samples, which effectively compensates the problem of insufficient training samples for few-shot charges, and we design a category prior loss function to further focus on few-shot charges;
(2) We creatively construct the charge label heterogeneous graph and a novel Dropedge graph convolutional network (DGCN) is designed to extract higher-order distinguishability features for effectively predicting few-shot and confusing charges;
(3) Extensive experiments are conducted on real-world datasets, and our model can effectively predict few-shot and confusing charges and outperforms other models.

The results show that our model outperforms all state-of-the-art methods.

2. Related Work

With the rapid development of artificial intelligence technology, some artificial intelligence applications such as intelligent justice [1,3–5] and intelligent transportation [11] have become research hotspots. Charge prediction is the key link of intelligent judicial trial. Firstly, our work is relevant to the existing few-shot and zero-shot classification. In recent years, some researchers have put forward solutions to this problem. For example, Sun et al. [12] proposed a hierarchical attention prototype network, designed feature-level, word-level and instance-level multiple cross-attentions, and made full use of the few-shot information. Zhang et al. [13] proposed a two-step framework that combines multiple semantic knowledges to solve the zero-shot text classification. Geng et al. [14] combined a dynamic routing algorithm with a meta-learning framework to simulate human generalization ability and effectively improve the classification effect of small samples. Bao et al. [15] used the meta-learning framework to classify low-frequency text, and leverage the distributional signatures of words, which encode pertinent word occurrence patterns. These works provide a good academic reference for the few-shot charge prediction task in this paper.

In addition, this paper works on the task of charge prediction in the LJP. For this task, many researchers have proposed different methods to improve the prediction effect. Dong et al. [16], for the multi-label charge prediction problem, using label co-occurrence, predicted multiple charges with article information on imbalanced data occasion. Chao et al. [17] proposed a deep neural framework to extract short but charge-decisive texts as the interpretation of charge prediction, which improved the interpretability of prediction results. Pan et al. [18] proposed a multi-scale attention model for different defendants and introduced Gaussian function into the model, effectively solving the problem of multiple defendants in charge prediction. Zhong et al. [19] followed two essential principles in LJP, Presumption of Innocence and Elemental Trial, and used the reinforcement learning method to extract the interpretation principles of cases so as to improve the interpretability

of predicted results. Yang et al. [20] designed a multi-perspective and bi-feedback neural network to analyze the LJP subtasks' dependence and predicted the charge with legal articles and imprisonment. However, only a few works focus on few-shot charges or confusing charges. The above methods are only good at predicting when the amount of data is sufficient, and the model performance decreases when the training data are unbalanced or the samples are small. Few-shot and confusing charge problems in charge prediction are also the focus of this study. Luo et al. [21] used legal articles as external knowledge and proposed a neural network with an attention mechanism to predict the charges, which improved the charge prediction results and provided legal bias. Hu et al. [9] constructed ten discriminative attributes of charges for few-charge prediction. Liu et al. [22] used a seq2seq model and designed an attention-based seq2seq model to predict the charge. Xu et al. [8] used a graph neural network to solve the confusing crime problem in LJP. However, the above methods do not deeply analyze its high-order representation of fact description, and the advantages are not obvious when there are few cases of some confusing charges. Therefore, it is necessary to enhance the data for low-frequency charges.

Cheng et al. [23] proposed a knowledge-aware model in which the graph convolution neural network is used to extract the fact description features, the legal schematic knowledge is regarded as a tree structure, and the self-attention network is used to model the hierarchical relationship. Inspired by this work, this paper introduces the charge graph information into the few-shot charge prediction to enhance the features, so as to further understand the external knowledge and fact description. The differences and improvements of our method to this literature are: (1) The research purposes are different. It mainly solves the easily confusing charges, and this paper improves the prediction effect of few-shot charges through feature augmentation while solving the easily confusing charges; (2) The modeling methods of external knowledge are different. In this paper, the charge label is modeled as multiple groups of heterogeneous graphs. We consider not only the hierarchical relationship between labels, but also the similar relationship between peer labels and the co-occurrence relationship between words in the label definition. Therefore, the distinguishing characteristics are more diverse. In addition, the improvement of loss function in this paper can effectively alleviate the imbalance of charge categories.

Therefore, the existing research on charge prediction mainly improves the model performance from the perspective of multi-task joint learning and the integration of external knowledge. Unlike the existing work, we focus on solving the problem of low-frequency and confusing charges from the perspective of internal text data augmentation and external charge label feature augmentation, which is obviously innovative. In addition, we use a layer of a feature fusion network to effectively incorporate the charge graph information into the fact representation, and the network is scalable when other external knowledge is incorporated.

3. Method

For each case, the fact description can be seen as a word sequence $X = \{x_1, x_2, \ldots, x_n\}$, where n represents the sequence length and $x_i \in V$, V is a fixed vocabulary. In addition, we extracted the charge label information C from FindLaw (https://china.findlaw.cn/zuiming/, accessed on 10 September 2021). Given the fact description X and charge label information C, the goal was to predict the charge $y \in Y$ from charge set Y in each case.

This paper proposed a charge prediction model to solve the few-shot and confusing charge problems in charge prediction. The overall structure of this model is shown in Figure 2, which mainly includes a data augmentation module and feature augmentation module. After encoding the distributed representation of cases, the Mixup method was used to generate virtual samples through linear interpolation. Our paper applied the algorithm proposed by Zhang et al. [10] in the field of image classification to the text classification in Section 3.1.2. Then, we modeled the charge labels as a heterogeneous graph in Section 3.2.1, and we designed a DGCN in Section 3.2.2. The DGCN was used to extract the distinguishability features, and the charge graph information was obtained. Then, a

feature fusion network was proposed in Section 3.3, which was used to effectively integrate the charge graph information into the case to learn semantic-enhanced fact representation. Finally, the semantic-enhanced fact representation was used to predict the charge. In addition, we constructed a category prior loss function, which pays more attention to low-frequency charge during the training process, thus improving the prediction effect of the model as a whole.

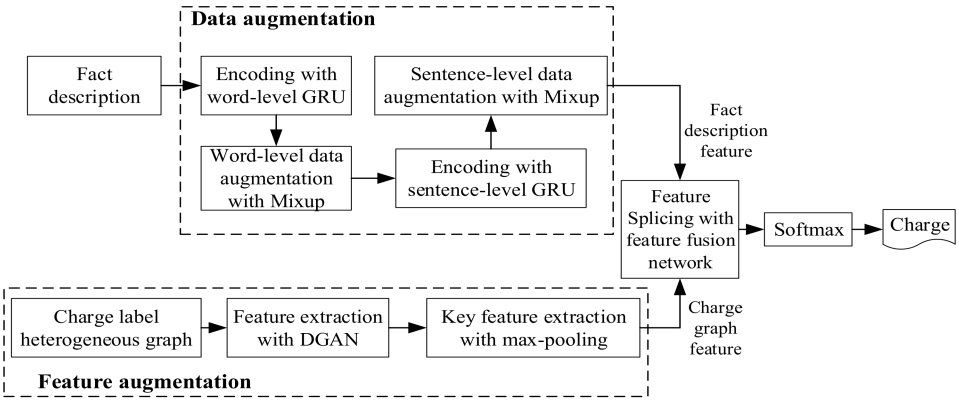

Figure 2. The structure of the charge prediction model.

3.1. Data Augmentation

In many few-shot classification tasks, fine-grained feature extraction or the introduction of external knowledge are often used to increase feature diversity, but sufficient data is still the premise to ensure the training effect of the model, so data augmentation is necessary. Data amplified by EDA [24] are single and limited, and data amplified by adding noisy are uncontrollable. Therefore, using limited cases to generate feature-diversified data is the key to data augmentation. Among many algorithms, the Mixup method can be used to cross-task the model training process, generate cross-label data, and, finally, obtain the augmented data with diverse features. In addition, the fact description is usually a hierarchical structure, so this paper used a word vector to represent the sentence and then a sentence vector to represent the whole fact description. The corresponding data augmentation also included two parts, word-level data augmentation and sentence-level data augmentation, and the data augmentation part of each level included encoder and data augmentation with Mixup.

3.1.1. Encoder

Given a sequence of fact descriptions X containing L sentences, we chose bi-directional gated recurrent unit (Bi-GRU) [25] to calculate the hidden state of words. Bi-GRU not only effectively solves the problems such as gradient disappearance of traditional RNN, which is widely used in the field of LJP for case coding, but also has a simple structure compared with pre-trained language models such as BERT [26] and is less prone to over-fitting in the few-shot classification task. Specifically, based on each sentence $S_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,m}\}$ (m represents the number of words contained in the $i-$th sentence and $x_{i,j}$ represents the $j-$th word in the $i-$th sentence) of fact description, we used a forward GRU and a reverse GRU to calculate the hidden states $h_{i,j}$.

$$h_{i,j} = [\overrightarrow{GRU}(e(x_{i,j})), \overleftarrow{GRU}(e(x_{i,j}))], j = 1, 2, \ldots, m, \quad (1)$$

where $h_{i,j} \in R^{2d_w}$ (d_w is the dimension of the hidden layer) denotes the hidden representation of the $j-$th word of the $i-$th sentence in the sequence and $e(;)$ is the word embedding code. This paper used a skip-gram algorithm of word2vec [27] to model the word vector. However, not all words in the fact description are equally important to the sentence expression, so an attention mechanism was used to extract words that are mean-

ingful to the sentence and calculate the weights $[\alpha_{i,1}, \alpha_{i,2}, \ldots, \alpha_{i,m}]$, where $\alpha_{i,j}$ is the weight of $x_{i,j}$. The combination of the attention mechanism can obtain deeper text features, so that Mixup can produce augmented data with diverse features. $\alpha_{i,j}$ is calculated as follows:

$$\alpha_{i,j} = \frac{\exp(\tanh(W_w h_{i,j})^T u_i)}{\sum_j \exp(\tanh(W_w h_{i,j})^T u_i)}, \tag{2}$$

where $W_w \in R^{d_w \times 2d_w}$ are trainable weight parameters and $u_i \in d^{d_w}$ is the word-level context vector. u_i was used to measure the importance of words generated by random initialization and jointly learned during the training process. Then, we obtained the charge information word-level hidden states $h'_{i,j} = \alpha_{i,j} h_{i,j}$.

3.1.2. Data Augmentation with Mixup

The main idea of the Mixup data enhancement algorithm proposed by Zhang et al. [10] in the field of image classification is to generate samples to amplify data by mixing two randomly extracted images and corresponding labels. Verma et al. [28] proposed the Manfold–Mixup method to generate pseudo-samples in embedding space.

$$\widetilde{x} = \lambda g_k(x_i) + (1-\lambda) g_k(x_j); \widetilde{y} = \lambda y_i + (1-\lambda) y_j, \tag{3}$$

where $g_k(\cdot)$ represents the forward process from input to the $k-$th layer in neural network coding, $\lambda \in [0,1]$ is the mixing factor, and (x_i, y_i) and (x_j, y_j) are a pair of samples randomly selected from the same batch. This paper refers to the idea of Verma et al. [28], as shown in Figure 3. We used interpolation in the information word hidden space to amplify data features, regularize the model to make it linear in the training data, and generate new states $\widetilde{h}_{i,\_} \in R^{2d_w}$ with rich features [29,30].

$$\widetilde{h}_{i,\_} = \lambda h'_{i,a} + (1-\lambda) h'_{i,b}, \tag{4}$$

where $\lambda \in [0,1]$ is the sample mixing factor, λ can be obtained by sampling the $Beta(\alpha, \alpha)$, α is a super parameter, $h'_{i,a}$ and $h'_{i,b}$ are hidden states of two words randomly selected from the $i-$th sentence, and a and b are the $a-$th and $b-$th words of the $i-$th sentence, respectively. $\widetilde{h}_{i,\_}$ is the new hidden state of the $i-$th sentence. Then we aggregated these state vectors of the $i-$th sentence to get sentence vectors $e_{s_i} = \sum_{j=1}^{m} \alpha_{i,j} \widetilde{h}_{i,j}$.

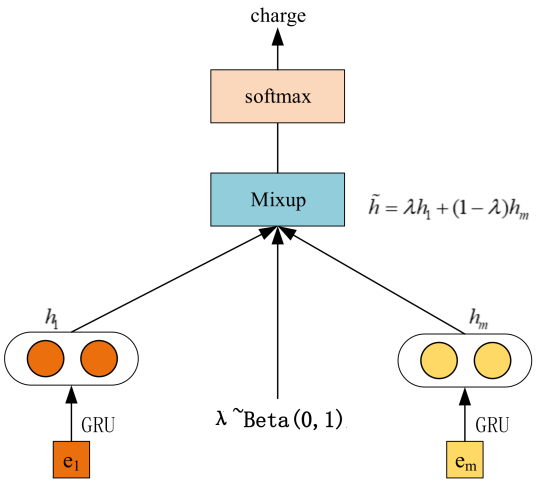

Figure 3. Data augmentation with Mixup.

Inspired by Yang et al. [31], who used a hierarchical attention mechanism for document classification, we attentively performed data augmentation based on word-level

Bi-GRU and sentence-level Bi-GRU. Therefore, the sequence of sentence representations $e_s = [e_{s_1}, e_{s_2}, \ldots, e_{s_L}]$ was obtained after word-level enhancement. Similarly, we used a sentence-level Bi-GRU to compute a sentence-level attention vector $\widetilde{h}_i = \alpha_i e_{s_i}$ and perform sentence-level Mixup data augmentation. The calculation process was as follows:

$$h_i = [\overrightarrow{GRU}(e_i), \overleftarrow{GRU}(e_i)], i = 1, 2, \ldots, L, \tag{5}$$

$$\alpha_i = \frac{\exp(\tanh(W_s h_i)^T u_{si})}{\sum_i \exp(\tanh(W_s h_i)^T u_{si})} \tag{6}$$

$$\widetilde{h}_{\_} = \lambda \widetilde{h}_a + (1 - \lambda) h_b \tag{7}$$

where α_i represents the sentence-level attention weight and $W_s \in R^{d_w \times 2d_w}$ is the weight matrix. We introduced a sentence-level context vector $u_{si} \in R^{d_w}$ and used the vector to measure the importance of the sentence. Finally, we obtained the fact representation $V = \sum_i \alpha_i \widetilde{h}_i$ after data augmentation.

3.2. Feature Augmentation

Realistic charge labels are often not independent structures, and each label not only has its basic definition and category, but also has similar or mutually exclusive relationships with other labels. In many existing studies of LJP tasks, the dependencies or correlations among the charge labels are usually ignored, and such dependencies can introduce additional valuable information for the charge prediction task, effectively solving the challenge of insufficient features for the few-shot task. At the same time, by further mining the differences between labels, it can provide distinguishable features for confusing charges. Therefore, it is necessary to conduct a deep analysis of the charge labels. In addition, the graph neural network has attracted a lot of attention due to its superior performance in graph-structured data [32], which can analyze the relationship between nodes and capture the features of node attributes with high accuracy at low cost compared with the fully connected multilayer perceptron (MLP). Therefore, we modeled the charge labels as graph structures, performed feature augmentation, and used a novel DGCN model to capture the charge label features.

3.2.1. Charge Label Heterogeneous Graph

As shown in Figure 1 above the structure of the charge label, each category includes a variety of charges (i.e., labels). Each label has a specific definition, which can be used for word segmentation. Therefore, this paper established two types of nodes V, word nodes and label nodes, and establishes edges E between nodes: word–word edge, word–label edge, and label–label edge. The structure is shown in Figure 4. On the right is the heterogeneous graph schema; on the left is a set of heterogeneous graphs with three labels and K words. Based on different categories, we constructed multiple groups of heterogeneous graphs. The total number of nodes in every graph is the number of labels plus the number of unique words charge information [33]. We considered every word and label as a one-hot vector as the input model and used the identity matrix as the feature matrix [34]. The label–word edges can be built based on word occurrence in labels, and the word–word edges can be built based on word co-occurrence in the whole corpus. Specifically, the weights between two word nodes can be calculated by point-wise mutual information (PMI) [35], which is a common method of word associations. Since the words are specific explanations of labels, the weight of the edge between a label node and a word node is the term frequency-inverse document frequency (TF-IDF) of the word in the label [36]. The same heterogeneous graph of the charges is relevant; we used similarity to

define the weights between two label nodes. Formally, we calculated the weights of edge between node i and node j as follows:

$$A_{ij} = \begin{cases} PMI(i,j) & i \text{ and } j \text{ are words} \\ TFIDF(i,j) & i \text{ is a label and } j \text{ is a word} \\ similarity(i,j) & i \text{ and } j \text{ are labels} \\ 1 & i = j \\ 0 & otherwise \end{cases} \tag{8}$$

where $PMI(i,j)$ is a word pair i, j and the PMI value is computed as $PMI(i,j) = \log \frac{p(i,j)}{p(i)p(j)}$, where $p(i,j) = \frac{W(i,j)}{W(i)}$, $p(i) = \frac{W(i)}{|W|}$. $|W|$ is the total number of sliding windows. $W(i)$ is the number of sliding windows containing word i in the corpus and $W(i,j)$ is the number of sliding windows containing both word i and j. PMI values reflect the semantic relevance between words; a positive PMI value indicates high semantic relevance and a negative value indicates low or no semantic relevance, so this paper only added edges between word pairs with positive PMI values. In addition, $TFIDF(i,j) = TF(i,j) * IDF(j)$, where $TF(i,j)$ is the number of times word appears in the label i, $IDF = \log(\frac{N_{all}}{N_j+1})$, N_{all} is the total number of tags, and $N_w + 1$ is the number of labels that contain the word j. We used the cosine similarity of the word vector as the weight between two label nodes, $similarity(i,j) = \frac{v_i \cdot v_j}{\|v_i v_j\|}$, where v_i and v_j are the vectors of label i and label j, respectively, and word2vec was used to calculate the word vector of the label.

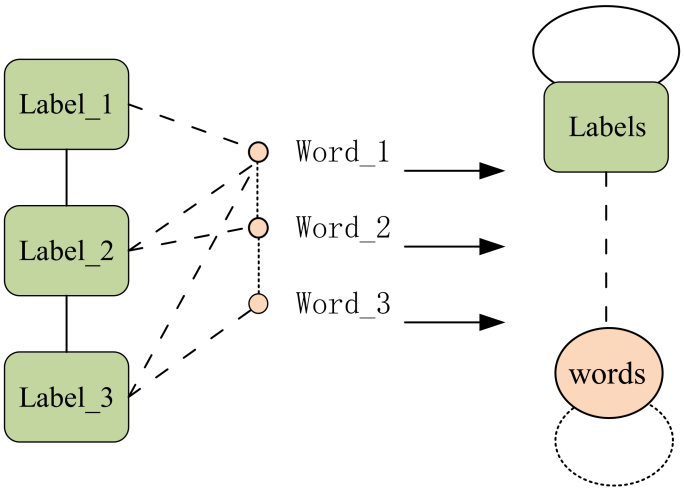

Figure 4. Charge label heterogeneous graph.

3.2.2. Dropedge Graph Convolutional Network

Based on the charge label heterogeneous graph established above, this paper selected the graph convolution network (GCN) for feature extraction. Compared with other graph neural network models, GCN has a simple structure and easy operation and can extract more effective features based on fewer parameters. The traditional GCN performs feature extraction on homogeneous graphs [37], while the charge label graph in this paper contains different entities and relationships, which are typical for heterogeneous graphs. By analyzing different types of information in the charge label graph, complex semantics and inter-word dependencies can be captured, which enriches the few-shot charge semantics more and effectively improves the accuracy of prediction results. Formally, consider a graph $G = (V, E)$, where $V(|V| = m)$ and E are sets of nodes and edges, respectively. Further, in this paper, the message passing is used to update the representation of each node. The graph convolution of the GCN model is actually a special form of Laplacian smoothing [38]. We obtained new features for each node by calculating the weighted

average of its own nodes and its neighbors. In addition, information from neighboring can be collected and integrated to update each node's own representation. The new d_g dimensional node feature matrix $H^{(l+1)} \in R^{m \times d_g}$ is calculated as follows:

$$H^{(l+1)} = \sigma(\widetilde{A}H^{(l)}W^{(l)}) \tag{9}$$

where $\widetilde{A} = \widetilde{D}^{-\frac{1}{2}}A'\widetilde{D}^{-\frac{1}{2}}$, $A' = A + I$ is the adjacency matrix A with added self-connection I and A is the weight matrix calculated by formula (8). $\widetilde{D}$ is a diagonal matrix with $\widetilde{D}_{ii} = \sum_j A'_{ij}$, $H^{(l)}$ as the node representation matrix of the $l-$th layer. $W^{(l)}$ is the trainable parameter matrix. $\sigma(.)$ is the activation function, such as ReLU. Initially, $H^{(0)} = X$.

In order to obtain higher-order charge information representation, the GCN contains 4 hidden layers. The distributed representation of each node in the last layer can be learned by applying the propagation rule to our graph defined before. However, while simply increasing the depth of the GCN network, the problem of over-smoothing arises and there is convergence in the node representation, thus making the differentiation of different types of nodes poor, which hinders the distinction of confusing charges. Therefore, we introduced the Dropedge [39] technique by randomly setting some non-zero elements of the adjacency matrix A to zero, i.e., deleting a fixed proportion of edges in the original graph, we called the new model Dropedge graph convolutional network (DGCN) and made the node connections more sparse without affecting the original features, avoiding the over-smoothing problem caused by increasing the number of GCN layers. At the same time, the Dropedge technique enhances the randomness and diversity of the data and alleviates the problem of over-fitting. If we denote the resulting adjacency matrix as A_{drop}, then its relation with A becomes:

$$A_{drop} = A - A' \tag{10}$$

where A' is a sparse matrix expanded by a random subset of non-zero elements of the adjacency matrix A. Then, we also performed the re-normalization trick on A_{drop}, leading to $\widetilde{A}_{drop}$. We replaced $\widetilde{A}$ with $\widetilde{A}_{drop}$. To further improve charge information' distinguishable feature, we computed the charge graph information $c = [c_1, c_2, .., c_k]$ by adopting two kinds of pooling operators to extract the distinguishing features of hidden states. The calculation is as follows:

$$c_i = W_c[\max(\{H_i\}_{i\in[1,k]}); avg(\{H_i\}_{i\in[1,k]})] \tag{11}$$

where $H_i \in R^{d_g}$ denotes hidden layer representation of the $i-$th heterogeneous graph, $i \in [1,k]$, k is the number of the heterogeneous graph, $W_c \in R^{d_g \times 2d_g}$ is the weight matrix, $[;]$ means concatenate operation, and $\max(\cdot)$ and $avg(\cdot)$ are max-pooling and avg-pooling operators, respectively.

3.3. Feature Fusion Network

Some feature splicing layers usually use simple feature splicing or feature summing without considering the relationship between external knowledge and internal descriptions, while we observed that the charge label features obtained by feature augmentation did not have exactly the same impact on the charge prediction task, and further analysis of the relationship between the charge label features and the fact descriptions could achieve a reasonable use of external knowledge. Therefore, we designed a layer of a feature fusion network to select more relevant information from the charge labels. As shown in Figure 5, formally, we defined an alignment matrix $U \in R^{n \times k}$ between charge label features and fact description features. Specifically, this paper first took charge graph information c through a layer of a fully connected network to get g as the hidden layer representation of the fusion network, then measured the importance of words through the fact description and

the similarity between labels, and obtained the standardized importance weight through softmax function. The specific formula is as follows:

$$g_i = \tanh(W_u c_i + b_u) \tag{12}$$

$$U_i = \frac{\exp(g_i{}^T W_v v)}{\sum_t \exp(g_i{}^T W_v v)} \tag{13}$$

where $W_u \in R^{d_w \times d_g}$ represents the weight matrix of the charge label feature and $W_v \in R^{d_w} \times d^{d_w}$ represents the weight matrix of fact description. Based on U, we computed the charge label embedding $e_c = (cU^T)$. Finally, we obtained the semantic-enhanced fact representation $f \in R^{d_w}$ by concatenating the fact description v with charge label embedding.

$$f = W_f[v; e_c] \tag{14}$$

where $W_f \in R^{d_w \times (d_w + d_g)}$ denotes weight matrix and $[;]$ means the concatenate operation.

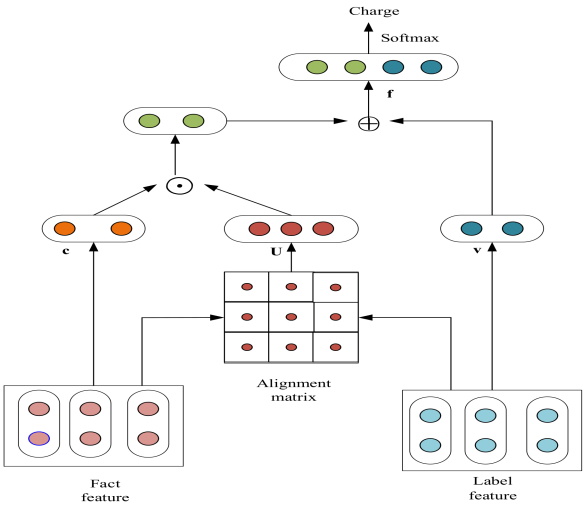

Figure 5. Feature fusion network.

3.4. Charge Prediction

The fact description after data enhancement and feature enhancement can effectively predict the few-shot charge and can better distinguish the confusing charges. Finally, we used the semantic-enhanced fact representation f from the feature fusion network to predict the final charge. Based on f, the $softmax$ function was used to get the prediction results. The specific calculations are as follows:

$$y = softmax(W_y f + b_y) \tag{15}$$

where $W_y \in R^{d^w \times d^w}$ and $b_y \in R^{d_w}$ are the weight matrix and bias in the output layer.

3.5. Optimization

Charge prediction is a typical text classification problem. We used cross entropy to optimize this task, and the loss function of charge prediction is as follows:

$$L = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \tag{16}$$

where C denotes the total number of classes, $\hat{y}_i$ denotes prediction probability, and y_i refers to the ground-truth label.

However, due to the imbalance of charge categories, the direct application of cross-entropy loss to the charge prediction task leads to the decline of prediction performance. Intuitively, in order to improve the training effect, higher weight should be given to the loss of low-frequency charges, and whether it is low-frequency depends on the proportion of a certain charge text in the dataset, that is, the prior distribution of charge categories. Therefore, it is crucial to fuse the prior distribution of the charges in the loss function. Inspired by Menon et al.'s [40] solution to the "long tail" problem, this paper further modeled the cross-entropy and category prior precedence as mutual information, i.e., the cross-entropy loss and the prior probability of the charge were combined to determine the final loss function. Specifically, the last step of charge prediction was softmax classification: we supposed the distribution of softmax classification layer input is $f(x;\theta)$, then the classification layer was generalized as $p_\theta(y|x) = \frac{e^{f_y(x;\theta)}}{\sum_{i=1}^{C} e^{f_i(x;\theta)}}$ and the prior probability distribution $p(y)$ of each charge. The specific combination rules are as follows:

$$\log \frac{p_\theta(y|x)}{p(y)} \sim f_y(x;\theta) \Leftrightarrow \log p_\theta(y|x) \sim f_y(x;\theta) + \log p(y) \tag{17}$$

Considering the classification layer function, softmax normalization was performed, as shown in Formula (18). Finally, a category prior loss function L_p was obtained.

$$p_\theta(y|x) = \frac{e^{f_y(x;\theta)+\log p(y)}}{\sum_{i=1}^{C} e^{f_i(x;\theta)+\log p(i)}} \tag{18}$$

$$\begin{aligned} L_p &= -\log p_\theta(y|x) \\ &= -\log \frac{e^{f_y(x;\theta)+\gamma \log p(y)}}{\sum_{i=1}^{C} e^{f_i(x;\theta)+\gamma \log p(i)}} \\ &= \log[1 + \sum_{i \neq y} (\frac{p(i)}{p(y)})^{\gamma} e^{f_i(x;\theta)-f_y(x;\theta)}] \end{aligned} \tag{19}$$

where γ denotes the regulatory factor.

4. Experiment

To evaluate the performance of the model in this paper, multiple sets of experiments were conducted on real datasets and compared with several state-of-the-art baseline models, as well as ablation experiments to demonstrate the validity of each module in the model. The hardware environment and software environment of the experiment are shown in Tables 1 and 2.

Table 1. Experimental hardware environment.

Hardware Name	Quantity	Parameter Description
CPU	1	Inter Core i7-4900MQ
Memory	2	Hynix Skhynix 32G
Graphics card	1	NVIDIA Tesla V100 32G
SSD	1	Samsung SAMSUNG 512G
Mechanical hard disk	1	Seagate 1TB

Table 2. Experimental software environment.

Software Name	Parameter Description
Operating system	Windows 10
Development tool	Pycharm 2020
Version of Python	Python 3.6.13
Version of Tensorflow	Tensorflow = 1.14.0
Version of Numpy	Numpy = 1.16.0

4.1. Dataset

We used the available dataset from [9], which contains real cases for few-shot charges prediction, and the dataset has three subsets of different sizes, denoted as Criminal-S (small), Criminal-M (medium), and Criminal-L (large). Each case has a good structure, including fact description, charge, related article, and prison term. This paper chooses the fact description part of the case as input and extracts the charge label by regular expression. Since this paper explores the few-shot prediction task, and the dataset contains cases with multiple defendants and multiple charges, which increase the complexity of the charge prediction task, we filtered these multi-label cases. We kept 149 distinct charges with at least 10 cases. The detailed statistics are shown in Table 3. All datasets were divided into training set, validation set, and test set in the ratio of 8:1:1. In addition, this paper counted the statistical results of the number of distribution of charges in Criminal-S. It can be seen from the Figure 6 that the distribution of charges was extremely unbalanced. More than 50% of the charges had less than 200 training data, while there were only 10 charges with more than 2000 training data.

Table 3. The statistic of different datasets.

Datasets	Criminal-S	Criminal-S	Criminal-L
Train	61,589	153,521	306,900
Validation	7755	19,250	38,429
Test	7702	19,189	38,429

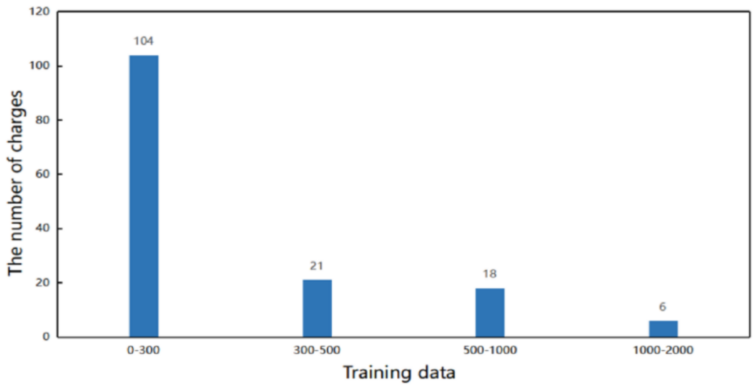

Figure 6. Imbalance graph of crime distribution.

The overall model was divided into two parts: data augmentation module and feature augmentation module. Tensorflow was uniformly used to build the neural network model. In the data augmentation module, in order to improve the data quality, we firstly cleaned the data and removed some invalid samples, then we discretized a large amount of information, normalizing the license plate number, mobile phone number, bank card number, and other information. The regular expression was used to extract the fact description from cases, THULAC was used for word separation, and the skip-gram algorithm of word2vec

for word vector modeling: setting the vector dimension to 200, setting the maximum text length to 500, truncating if it was too long, and filling it with 0 if it was less than 500. The maximum document length was 32 sentences, words with a word frequency of less than five were regarded as unknown words, and the dimension of GRU hidden layer d_w was set to 256. The dropout was set to 0.5, the mixing factor λ was obtained from the beta distribution ($Beta(\alpha, \alpha)$) sampling, the parameter was $\alpha = 150$, the batch size was 64, and the learning rate was 0.01.

In the feature augmentation module, the charge definition was also segmented, the word vector was modeled by using the skip-gram algorithm of word2vec; the vector dimension was set to 200, the window size was set as 20, and the dropout to 0.5. All neural network models were trained by random gradient descent (SGD), and Adam [41] was used as the optimizer. Iterative training was repeated during the experiment until the difference between two consecutive iterations was small enough, and the maximum epochs were 20. We employed accuracy (Acc.), macro-precision (MP), macro-recall (MR), and macro-F1 (F1) as our evaluation metrics.

4.2. Baselines

We selected a basic neural network model and several state-of-the-art models of LJP as a baseline. The effectiveness of our model is proved by comparative analysis. The descriptions of the baseline models are as follows:

CNN [42]: A text classification model based on CNN. Text features are extracted using CNNs with different width filters, combined with fully connected neural networks to predict the charges.

HAN [31]: A hierarchical attention network model for document classification. The model uses word-level GRU to extract features, an attention mechanism to aggregate words into sentences and further extract features from sentence-level GRU, and an attention mechanism to classify documents.

Few-attribute [9]: A model for few-shot charges prediction. The model introduces legal attributes as external knowledge, and multi-task joint modeling is used to predict both charges and attributes.

MPBFN [20]: A multi-perspective bi-feedback network for legal judgment prediction. The network effectively utilizes the dependencies among subtasks of LJP, and a word collection attention mechanism is proposed to increase the accuracy of penalty prediction.

LADAN [8]: A model for confusing charges prediction. The model uses a graph neural network to extract the differences between legal articles and re-encode the case descriptions to predict the charges.

In the experimental process, this paper treated the charge prediction as a classification problem, selected a set of optimal experimental parameters by reviewing the range of experimental parameters given in the original paper, conducted three iterations of the experiment, and took the best single result as the final result.

4.3. Results and Analysis

The experimental results on the criminal dataset are shown in Table 4. † indicates significant improvements ($p < 0.05$) compared with the best baseline. It can be observed that our model achieves better performance on three different-sized datasets, which showcases the robustness and wide applicability over various application scenarios of our proposed method for charge prediction. Our model is basically similar to other models on a certain metric of a dataset, such as the Acc (93.3) of LADAN and ours (93.5) are close to each other in Criminal-S. Through our annotation, it can be found that at least two indicators of our model are significantly improved in different datasets, especially in MR and F1. Thus, we can infer that the model in this paper significantly outperforms other baseline models on different datasets by looking at all the metrics for a particular dataset. In addition, the experimental results of different-sized datasets are compared horizontally, and it is found that the prediction results become more and more accurate with the increase of data

volume, which proves that the amount of data can affect the training effect of the model, and the training data of large datasets are more adequate.

Table 4. Charge prediction results of three datasets.

Datasets	**Criminal-S**				**Criminal-M**				**Criminal-L**			
Metrics	**Acc.**	**MP**	**MR**	**F1**	**Acc.**	**MP**	**MR**	**F1**	**Acc.**	**MP**	**MR**	**F1**
CNN	91.9	50.5	44.9	43.5	93.5	57.6	48.1	50.5	93.9	66.0	50.3	54.7
HAN	92.2	52.7	49.2	44.9	93.7	61.1	55.3	54.6	95.1	68.2	57.3	58.6
Few-attribute	92.8	57.0	53.9	53.4	94.7	66.7	60.4	61.8	95.7	73.3	67.1	68.6
MPBFN	93.3	63.4	57.9	56.2	94.7	69.1	68.2	66.9	95.8	76.6	73.6	72.5
LADAN	93.3	63.7	57.2	56.6	**94.9**	69.7	67.2	67.1	95.5	76.8	72.9	73.1
Our model	**93.5**	**66.2** †	**63.8** †	**62.1** †	**94.9**	**70.3** †	**73.5** †	**70.0** †	**95.9**	**78.6** †	**77.1** †	**76.2** †

Further comparing the experimental results, it can be observed that our model outperforms all the baseline models, and the existing models generally perform poorly on the F1 metric. This is mainly because of the imbalance of training samples among different charges and indicates the shortage of prediction for few-shot charges. In contrast, our model achieves an absolute improvement of 5.5%, 2.9%, and 3.1% in F1 over the current optimal model on the three datasets, respectively. It proves the effectiveness and robustness of the method in this paper under different scenarios (sufficient or insufficient data).

Among these evaluation metrics, MR and F1 are the preferred evaluation metrics for classification problems, especially in the scenario of unbalanced data. The above analysis shows that our model is superior in F1. Therefore, we compared the performance with MR metric, and further compared the generalization performance of our model and the baseline model in a statistical sense. Firstly, we compared six different models on three datasets, and then sorted them on each dataset according to MR values and assigned ordinal values (1, 2, ...). As shown in Table 5, the last row is average ordinal values. Then, the "Friedman test" was used to judge whether the performance of these models was the same. Let r_i be the average ordinal values of the $i-$th models. We assumed a variable $r_{\chi^2} = \frac{k-1}{k} \cdot \frac{12N}{k^2-1} \sum_{i=1}^{k} (r_i - \frac{k+1}{2})^2$, where $k = 6$ is the number of models and $N = 3$ is the number of datasets. We obtained the $r_{\chi^2} = 14.619$ when k and N are large and r_{χ^2} obeys distribution χ^2. Then, we used the variable $r_F = \frac{(N-1)r_{\chi^2}}{N(k-1)-r_{\chi^2}}$, which follows the F distribution with degrees of freedom $(k-1)$ and $(k-1)(N-1)$. Then, we obtained the $r_F = 76.740$ by referring to the table of common critical values for the F test, when significance level $\alpha = 0.05$, $r_F > 3.217$. Therefore, the assumption that all models have the same performance was rejected. Then we used the "Nemenyi follow-up test" by referring the common critical values for the "Nemenyi test". We obtained the critical value $q_\alpha = 2.850$ when $k = 6$ and $\alpha = 0.05$. The "Nemenyi test" calculates the critical range $CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} = 3.522$ of the difference between the average ordinal values. Finally, we compared the average ordinal values gap and CD:$6-1 > CD; 5-1 > CD$. This shows that, compared with the ordinary neural network model (CNN and HAN), the performance of our model is particularly superior. In addition, compared with other legal judgment prediction models, our model predicts a higher MR value, which is slightly better than other models in statistical sense.

Table 5. The ordinal values of different models.

Datasets	CNN	HAN	Few-Attribute	MPBFN	LADAN	Our Model
Criminal-S	6	5	4	3	2	1
Criminal-M	6	5	4	2	3	1
Criminal-L	6	5	4	2	3	1
Average ordinal value	6	5	4	2.33	2.667	1

By comparing the ordinary neural network model and the charge prediction model, it is found that the charge prediction model is significantly better than the ordinary neural network in various indexes, which indicates that in-depth analysis of various factors affecting LJP can effectively improve the prediction effect by introducing external knowledge or multi-task joint learning.

Comparing on three different size datasets, the advantage of our model is particularly obvious on the Criminal-S dataset, mainly due to the adoption of the Mixup method in this paper. The two most basic metrics in the classification domain are Acc. and MR, and, comparing our model with LADAN and HAN, the accuracy is basically close, but the recall differs greatly. The main reason is that low-frequency and confusing charges account for a lower proportion, while the LADAN model can effectively distinguish confusing charges through DGCN, so the MR is improved. Through data enhancement and feature enhancement, our model can not only distinguish easily confused charges, but also has a good prediction effect on few-shot charges.

To further demonstrate the advantage of our model in the case of few-shot charges, MPBFN and LADAN models, which are more effective in the baseline model, were compared with our model under different frequency charges. Here, according to the frequency of the charge, we divided them into three parts: the number of cases included in the low-frequency charges is less than 10 and the number of cases included in the high-frequency charges is more than 100. As shown in Table 6, the effects of the three models in high-frequency charges are close, but compared with the baseline model, our model improves 3.95% and 4.74% in low-frequency charges, respectively, which fully proves the advantages of our model in predicting few-shot charges.

Table 6. Macro-F1 values of various charges on Criminal-S.

Charge Type	Low	Medium	High
Charge number	49	51	49
MPBFN	53.1	61.6	82.8
LADAN	52.7	62.9	83.3
Our model	55.2	64.1	83.9

In addition, it can be seen from Figure 6 that the charge distribution of data in the legal field is imbalanced. In order to prove the effectiveness of this model in an unbalanced data scenario, we select a new evaluation metrics for comparative experiments. As shown in Table 4, in the ordinary neural network model the prediction effect of HAN is better than the CNN. In the LJP models, the Few-attribute model is specially used to solve the few-shot charge problem. Therefore, we selected HAN and Few-attribute from the baseline model. At the same time, the loss function of our model was replaced by the ordinary cross entropy; the variant is named cross-loss. We compared the above three models with our model on the Criminal-L set with sufficient data. Then the AUC evaluation index was used to explore the effectiveness of the model in the unbalanced data scenario. The experimental results are shown in the Table 7. It can be seen from the table that the AUC value of HAN is the lowest, which indicates the limitation of the ordinary neural network model for the category imbalance problem, and the accuracy can be effectively improved by integrating external

knowledge. Therefore, the prediction result of Few-attribute is significantly improved, and the AUC value of our model is the highest. When the ordinary cross-entropy loss function is used, the AUC value decreases significantly, but it is still higher than other models. It can be seen that the data augmentation and feature augmentation strategies in this paper can initially solve the problem of category imbalance. More importantly, the fusion of category priors has a significant effect on solving the problem of unbalanced data.

Table 7. AUC values of different models on Criminal-L.

Model	AUC
HAN	61.3
Few-attribute	73.9
Cross-loss	74.6
Ours	79.7

Further, we selected "theft" and "fire crime" as the representative of high-frequency charges and low-frequency charges respectively and used a PR curve (precision-recall) to intuitively explore the performance of our model and Few-attribute on different frequency charges. As shown in Figure 7, it can be observed that the prediction effect of the model on "theft" is higher than that on "fire crime", and, in the prediction of "theft", the prediction effects of the two models are basically close. This shows that the two models have the same prediction effect on high-frequency charges and also reflects that low-frequency charges in the dataset can affect the effect of the model. However, for low-frequency charges, the PR curve of our model completely covers another one, so our model is obviously better than the Few-attribute model. It can be seen that the method proposed in this paper is particularly excellent for few-shot charges.

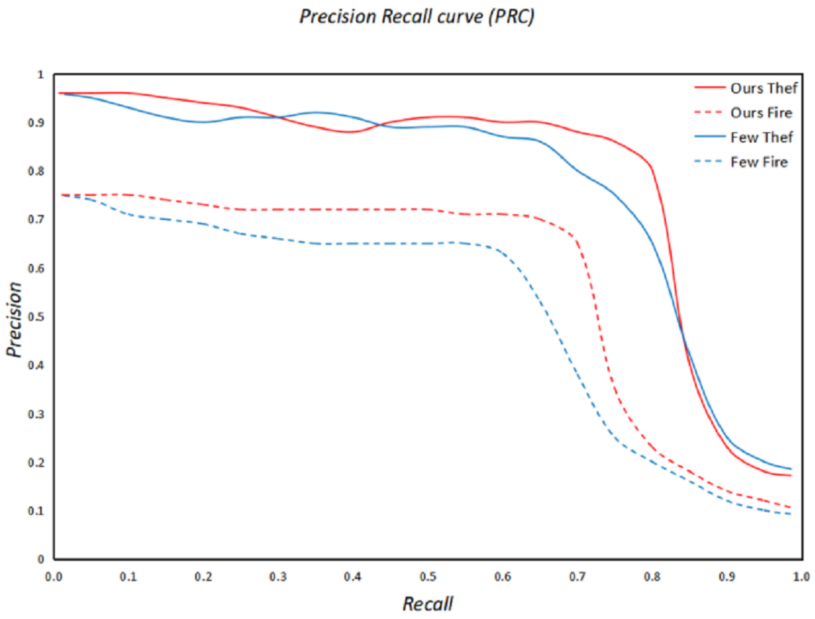

Figure 7. Performance comparison of our model and Few-attribute on specific charges.

4.4. Ablation Test

Our model is characterized by data augmentation and feature augmentation. Therefore, we designed an ablation test to verify the advantages of each module. Firstly, the data augmentation module was removed from the model, and the model only extracted the fact description features from word and sentence levels, and the variant model was named no-data. Subsequently, the feature augmentation module based on DGCN was removed,

and the model did not introduce external knowledge such as charge information, and the variant was named no-feature. Finally, the category prior loss function was replaced by the ordinary cross-entropy loss function, and the variant was named cross-loss. Experiments were conducted on three datasets and the results are shown in Table 8. It is observed that the prediction performance decreases when any of the modules is removed. Specifically, when the Mixup data augmentation module is removed, the performance decreases significantly on the Criminal-S dataset because the training samples can be implicitly augmented by Mixup strategy, thus improving the model's training effect. When the feature augmentation module is removed, the model performance decreases significantly because this paper extracts features from the charge information into the model by DGCN, which provides external knowledge for the low-frequency and confusing charges existing in the dataset, thus effectively improving the representation capability of the model. Finally, when ordinary cross-entropy loss is used instead of the optimized loss in this paper, the accuracy is slightly reduced, but the F1 value decreases significantly. By analyzing the results, the decrease in performance mainly originates from the performance on low-frequency charges, which accounts for a low proportion of the dataset, so the accuracy rate does not fluctuate much, further proving the effectiveness of the category prior loss function on few-shot charges.

Table 8. Macro-F1 values of various charges on Criminal-S.

Datasets	**Criminal-S**				**Criminal-M**				**Criminal-L**			
Metrics	**Acc.**	**MP**	**MR**	**F1**	**Acc.**	**MP**	**MR**	**F1**	**Acc.**	**MP**	**MR**	**F1**
No-data	91.1	60.4	59.7	58.4	92.9	67.2	71.3	68.2	95.5	76.2	76.0	74.9
No-feature	93.1	60.7	61.2	60.3	93.8	67.1	71.9	68.1	95.1	75.1	75.8	74.2
Cross-loss	92.4	59.7	60.2	59.1	93.6	68.1	71.3	68.7	94.9	77.3	76.2	75.1
Ours	93.5	66.2	63.8	62.1	94.9	70.3	73.5	70.0	95.9	78.6	77.1	76.2

4.5. Case Study

As shown in Figure 8, a representative case was chosen to illustrate visually how our model can improve the accuracy of prediction results. In this case, the defendant is charged with robbery. Theft and robbery are related to property, and their fact descriptions are similar, so many models have difficulty distinguishing them. In-depth analysis of the relevant legal articles reveals that one important difference between them is robbery often uses violence to seize the victim's property, and the crime of theft is mainly related to "taking advantage of a person's surprise" and "secret manner". In the charge prediction, the model can only correctly predict the charge when the distinguishable features in the fact description are extracted. As shown in Table 9, the charge of the case is accurately predicted by our model. On the contrary, other models did not focus on both "property" and "violence" and incorrectly predicted the crime of theft. For example, in MPBFN, by analyzing the relationship between the LJP subtasks and using the legal article and prison term to assist in the prediction of the charge, the case characteristics are extracted in depth; however, the information of "hit" and "injury" in the case is not paid extra attention to. However, in this paper, by constructing a heterogeneous graph of the charge information and using a DGCN to extract the heterogeneous graph features, our model can effectively capture the semantics related to the specific charge, which are assigned the red color in Figure 8, and, finally, combine external knowledge and fact description to accurately predict the charge in this case, which further proves that the feature augmentation based on graph attention network can effectively distinguish confusing charges.

Case-Robbery:

2016年5月8日21时30分许，被告人陈某某在光明新区驾驶摩托车撞倒被害人段某驾驶的电动自行车并致其受伤，同时抢走其手袋（内有苹果手机1部、现金1400元及段某的个人物品）。经鉴定，被抢手机价值4537元。

At about 21:30 on May 8, 2016, the defendant Chen drove a motorcycle in Guangming New Area, knocked down the electric bicycle driven by the victim Duan and injured him. At the same time, he robbed his handbag (containing an Apple phone, 1400 yuan in cash and Duan's personal belongings). After identification, the robbed mobile phone is worth 4537 yuan.

Figure 8. The case of robbery.

Table 9. Charge prediction results of robbery case.

Models	Ours	CNN	HAN	Few-Attribute	MPBFN
Result	Robbery	Theft	Theft	Theft	Theft

5. Conclusions

This paper solves the problem of few-shot charges in charge prediction by means of data augmentation and feature augmentation and proposes a category prior loss function to further alleviate the difficulty of inadequate training for few-shot charge cases, thus improving the prediction effect of the model overall. Extensive experimental results show that our model significantly improves on all indicators of criminal cases, and the ablation experimental results prove that the virtual samples are effectively generated for few-shot charges through the Mixup data augmentation. Building heterogeneous graphs for feature augmentation based on charge information can introduce external knowledge and the prediction effect of few-shot charges is further improved. Moreover, the DGCN proposed in this paper can extract distinguishable features for confusing charges. In addition, a feature fusion network is used to learn semantic-enhanced fact representation, and the network has good scalability for the introduction of other external knowledge. In the future, solutions will be proposed for complex situations such as multiple defendants and multiple charges.

Author Contributions: Conceptualization, P.W.; Data curation, P.W.; Formal analysis, P.W.; Funding acquisition, X.Z.; Investigation, Z.C.; Methodology, P.W.; Software, P.W.; Supervision, X.Z.; Validation, Z.C.; Visualization, X.Z.; Writing—original draft, P.W.; Writing—review and editing, X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Key R&D Program of China (Grant No. 2018YFB1601502), the LiaoNing Revitalization Talents Program (Grant No. XLYC1902071) and the Fundamental Research Funds for the Central Universities (Grant No. 3132019313).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhang, H.; Dou, Z.C.; Zhu, Y.T.; Wen, J.R. Few-Shot Charge Prediction with Multi-grained Features and Mutual Information. In Proceedings of the 20th Chinese National Conference on Computational Linguistics, Huhhot, China, 22–24 October 2021; pp. 387–403.
2. Shen, Y.; Sun, J.; Li, X.P.; Zhang, L.; Shen, X.J. Legal Article-Aware End-To-End Memory Network for Charge Prediction. In Proceedings of the 2nd International Conference on Computer Science and Application Engineering, Huhhot, China, 22–24 October 2018; pp. 1–5.
3. Lin, W.C.; Kuo, T.T.; Chang, T.J. Exploiting machine learning models for chinese legal documents labeling, case classification, and sentencing prediction. In Proceedings of the ROCLING, Chung-Li, Taiwan, 21–22 September 2012; pp. 140–141.
4. Lauderdale, B.E.; Clark, T.S. The Supreme Court's Many Median Justices. *Am. Polit. Sci. Rev.* **2012**, *106*, 847–866. [CrossRef]
5. Katz, D.M.; Bommarito, M.J.; Blackman, J. A general approach for predicting the behavior of the supreme court of the united states. *PLoS ONE* **2017**, *12*, e174698. [CrossRef] [PubMed]
6. He, C.; Peng, L.; Le, Y.; He, J.; Zhu, X. SECaps: A Sequence Enhanced Capsule Model for Charge Prediction. In Proceedings of the International Conference on Artificial Neural Networks, Munich, Germany, 17–19 September 2019; Springer: Cham, Switzerland, 2019; pp. 227–239.
7. Paka, W.S.; Bansal, R.; Kaushik, A. Cross-SEAN: A cross-stitch semi-supervised neural attention model for COVID-19 fake news detection. *Appl. Soft. Comput.* **2021**, *107*, 107393. [CrossRef]
8. Xu, N.; Wang, P.; Chen, L. Distinguish Confusing Law Articles for Legal Judgment Prediction. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 3086–3095.
9. Hu, Z.; Li, X.; Tu, C. Few-shot charge prediction with discriminative legal attributes. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–26 August 2018; pp. 487–498.
10. Zhang, H.Y.; Cisse, M. mixup: Beyond empirical risk minimization. In Proceedings of the 6th International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
11. Zhou, Z.; Zhang, Y.; Wang, S. A Coordination System between Decision Making and Controlling for Autonomous Collision Avoidance of Large Intelligent Ships. *J. Mar. Sci. Eng.* **2021**, *9*, 1202. [CrossRef]
12. Sun, S.; Sun, Q.; Zhou, K. Hierarchical attention prototypical networks for few-shot text classification. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 5 November 2019; pp. 476–485.
13. Zhang, J.; Guo, P.Y. Integrating Semantic Knowledge to Tackle Zero-shot Text Classification. In Proceedings of the North American Chapter of the Association for Computational Linguistics, Minneapolis, MN, USA, 2–7 June 2019; pp. 1031–1040.
14. Geng, R.; Li, B.; Li, Y.; Ye, Y.; Jian, P.; Sun, J. Few-shot text classification with induction network. *arXiv* **2019**, arXiv:1902.10482.
15. Bao, Y.; Wu, M.; Chang, S. Few-shot text classification with distributional signatures. In Proceedings of the 8th International Conference on Learning Representations, ICLR, Addis Ababa, Ethiopia, 26–30 April 2020.
16. Dong, H.; Yang., F.; Wang., X. Multi-label charge predictions leveraging label co-occurrence in imbalanced data scenario. *Soft. Comput.* **2020**, *24*, 17821–17846. [CrossRef]
17. Chao, W.H.; Jiang, X.; Luo, Z.C. Interpretable Charge Prediction for Criminal Cases with Dynamic Rationale Attention. *J Artif. Intell. Res.* **2019**, *66*, 743–764. [CrossRef]
18. Pan, S.; Lu, T.; Gu, N. Charge Prediction for Multi-defendant Cases with Multi-scale Attention. In Proceedings of the Computer Supported Cooperative Work and Social Computing—14th CCF Conference, Kunming, China, 16–18 August 2019; pp. 766–777.
19. Zhong, H.; Wang, Y.; Tu, C. Iteratively Questioning and Answering for Interpretable Legal Judgment Prediction. In Proceedings of the Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 1250–1257.
20. Yang, W.M.; Jia, W.J.; Zhou, X.J.; Luo, Y.T. Legal Judgment Prediction via Multi-Perspective Bi-Feedback Network. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, Macao, China, 10–16 August 2019; pp. 4085–4091.
21. Luo, B.; Feng, Y.; Xu, J. Learning to Predict Charges for Criminal Cases with Legal Basis. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP, Copenhagen, Denmark, 9–11 September 2017; pp. 2727–2736.
22. Liu, Z.; Tu, C.; Sun, M. Legal Cause Prediction with Inner Descriptions and Outer Hierarchies. In Proceedings of the Chinese Computational Linguistics—18th China National Conference, Kunming, China, 18–20 October 2019; pp. 573–586.
23. Cheng, X.; Bi, S.; Qi, G. Knowledge-aware Method for Confusing Charge Prediction. In Proceedings of the Natural Language Processing and Chinese Computing—9th CCF International Conference, Zhengzhou, China, 14–18 October 2020; pp. 667–679.
24. Wei, J.; Zou, K. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP, Hong Kong, China, 3–7 November 2019; pp. 6381–6387.
25. Cho, K.; Merrienboer, B.V. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014; pp. 1724–1734.
26. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the NAACL-HLT, Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.

27. Mikolov, T.; Sutskever, I.; Chen, K. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of the 26th Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–8 December 2013; pp. 3111–3119.
28. Verma, V.; Lamb, A.; Beckham, C.; Najafi, A. Manifold Mixup: Better Representations by Interpolating Hidden States. In Proceedings of the ICML, Long Beach, CA, USA, 9–15 June 2019; pp. 6438–6447.
29. Amit, J.; Dwaraknath, G.; Ramit, S. Leveraging BERT with Mixup for Sentence Classification. In Proceedings of the Thir-ty-Fourth AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 13829–13830.
30. Chen, J.; Yang, Z.; Yang, D. MixText: Linguistically-Informed Interpolation of Hidden Space for Semi-Supervised Text Classification. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), Online, 5–10 July 2020; pp. 2147–2157.
31. Yang, Z.; Yang, D.; Dyer, C. 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, San Diego, CA, USA, 12–17 June 2016; pp. 1480–1489.
32. Hu, L.; Yang, T.; Shi, C. Heterogeneous Graph Attention Networks for Semi-supervised Short Text Classification. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China, 3–7 November 2019; pp. 4820–4829.
33. Chen, R.Y.; Yao, H.; Li, R.J.; Kang, X.J.; Li, S.W. Identifying Human Daily Activity Types with Time-Aware Interactions. *Appl. Sci.* **2020**, *10*, 8922. [CrossRef]
34. Wei, Z.Y.; Gui, Z.P.; Zhang, M.; Yang, Z.L.; Yu, J. Text GCN-SW-KNN: A novel collaborative training multi-label classification method for WMS application themes by considering geographic semantics. *Big Earth Data* **2020**, *10*, 1–24. [CrossRef]
35. Yao, L.; Mao, C.; Luo, Y. Graph Convolutional Networks for Text Classification. In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, AAAI, Honolulu, HI, USA, 27 January 27–1 February 2019; pp. 7370–7377.
36. Yang, Y.; Wu, B.; Li, L.W.; Wang, S.Y. A Joint Model for Aspect-Category Sentiment Analysis with TextGCN and Bi-GRU. In Proceedings of the 5th IEEE International Conference on Data Science in Cyberspace, Hong Kong, China, 27–30 July 2020; pp. 156–163.
37. Kip, F.T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In Proceedings of the 5th International Conference on Learning Representations, Toulon, France, 24–26 April 2017.
38. Li, Q.; Han, Z.; Wu, X.M. Deeper insights into graph convolutional networks for semi-supervised learning. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, LA, USA, 2–7 February 2018; pp. 3538–3545.
39. Rong, Y.; Huang, W.; Xu, T. DropEdge: Towards Deep Graph Convolutional Networks on Node Classification. In Proceedings of the 8th International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
40. Menon, A.K.; Jayasumana, S.; Rawat, A.S. Long-tail learning via logit adjustment. In Proceedings of the 9th International Conference on Learning Representations, Virtual Event, Austria, 3–7 May 2021.
41. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
42. Yoon, K. Convolutional neural networks for sentence classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014; pp. 1746–1751.