

## Research Article

# Application of Deep Learning Technology in the Recommendation System of Constitutional and Constitutional Cases

Zhongbo Wu <sup>1</sup>, Yuan Lu,<sup>1</sup> and Zijun Feng<sup>2</sup>

<sup>1</sup>School of Marxism, Qiannan Normal University for Nationalities, Duyun, 558000 Guizhou, China

<sup>2</sup>School of Information Science and Technology, Northeast Normal University, Changchun, 130012 Jilin, China

Correspondence should be addressed to Zhongbo Wu; wuzhongbo@sgmtu.edu.cn

Received 11 July 2022; Revised 9 August 2022; Accepted 25 August 2022; Published 21 September 2022

Academic Editor: Kuruva Lakshmana

Copyright © 2022 Zhongbo Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Due to the huge system in the field of constitution, the content of cases is complicated, and the number of documents related to constitutional cases is huge. It is incredibly tough to find the information that suits your demands in such a large volume of constitutional documents. The artificial neural network-based deep learning method has made significant progress. Deep learning has been used in the disciplines of computer image and video processing, as well as natural language processing, and has outperformed typical machine learning methods. To that purpose, the content-based recommendation algorithm in this work is improved using deep learning approaches. This paper primarily covers the following topics: (1) the current research status of deep learning technology in recommender systems is reviewed, and the theoretical knowledge of deep neural networks is introduced. (2) A text classification model OCCNN based on character-level convolutional neural network is proposed to solve the problem of document classification of massive cases. A semantic similarity calculation model TF-W2V based on word frequency and word vector is also proposed to solve the problem of semantic similarity calculation of cases. (3) The two methods are tested, and the experimental results show that the accuracy is much higher than that of the conventional method. The model can recommend cases that meet the needs of users according to the requirements put forward by users in the process of handling cases.

## 1. Introduction

According to the survey report of the International Data Corporation (IDC), the total global production data was about 0.49ZB in 2008, about 0.8ZB in 2009, increased to 1.2ZB in 2010, and 1.82ZB in 2011, which is equivalent to the global average per capita people generate more than 200GB of data. The research report given by IBM also pointed out that in the past two years, the data created by humans has reached 90% of all the data since human civilization, and by 2020, the data created by the world has accumulated; the total scale has exceeded 40ZB [1, 2]. In the era of big data, the scale of information has exploded, and the processing capabilities of the past have become inadequate. Taking the rapid growth of commodity types as an example, some scholars used the concept of The Long Tail for the first

time in Wired magazine in 2004 to describe that the products sold by websites such as Amazon and Netflix are in the state of The Long Tail develop. In this case, some commodities are sold very little, but they will not drop to zero, and it is a business and economic model with a much longer tail than the head of the curve, and they believe that this “long tail” effect stems from globalization, efficient supply chains, and individual demands. For the diversification and abundance of such information, too much information will prevent users from correctly locating valid data or cause the flood of invalid data, resulting in a serious low data utilization rate. In order to solve this problem, the recommendation algorithm was born. In short, it calculates the user’s preferences through the user’s past operation data and then recommends possible matching data for the user. Take news platforms as an example. In the past, it was mainly

represented by centralized content platforms such as Tencent News. Now, news applications represented by Toutiao push customized news with different content on the homepage according to user preferences, which has promoted the transformation of the entire industry to personalized recommendation. On the home pages of e-commerce websites such as Taobao, JD.com, and NetEase, there is a “Guess You Like” section, which recommends products for users based on their browsing history and purchase records. According to the Center for Data Science, for e-commerce sites like Amazon, recommender systems can take as much as 10% to 25% of incremental revenue. Current research on recommender systems also has problems to be solved, such as information sparse, cold start, and the need for a large amount of accurate item feature information. The combination of new technologies and recommendation algorithms may help recommender systems solve some problems [1]. In recent years, deep learning has achieved more results in computer graphics and images, natural language processing, etc., which has re-inspired people’s exploration and enthusiasm for deep learning. Deep learning has also become a more popular branch in the field of machine learning algorithms. In addition, major domestic and foreign companies have increased their research and investment in deep learning, and many companies have achieved fruitful results in the research and development of artificial intelligence commercial products. Papers on recommendation algorithms based on deep learning are also increasing year by year [3]. In 2012, Google officially released the knowledge graph project. Knowledge graphs can provide generic or domain knowledge to help other projects enrich information. In the research of recommendation algorithm, the same algorithm is applied in different fields, and its performance is not the same. By providing domain knowledge and helping models understand semantic information, knowledge graphs can make up for the unsuitable problem of recommendation models in a specific field, help improve recommendation accuracy, and have important research value. To sum up, the main work of this paper is to combine the recommendation system with deep learning technology, apply the improved recommendation algorithm to the field of constitution, and realize the recommendation system for constitutional and constitutional cases based on deep learning. This paper proposes a text classification processing model based on deep learning and a similarity calculation model. The content-based recommendation model is improved, and a judicial case recommendation system is implemented. The model can understand user semantics and provide users with accurate case recommendation services [4].

The unique contribution of the paper includes the following:

- (i) Exhaustive review of the current research status of deep learning technology-based implementations in recommender systems
- (ii) Development of a text classification model named as OCCNN based on character-level convolutional

neural network which proposes to solve the problem of document classification of massive cases

- (iii) Proposal of semantic similarity calculation model TF-W2V based on word frequency and word vector to resolve the problem of semantic similarity calculation of cases
- (iv) Testing and evaluation of the proposed approaches in comparison to the conventional methods

## 2. Related Work

The concept of deep learning was proposed by Hinton et al., who also proposed the deep structure of multilayer autoencoders [5]. In the early days, the performance of neural networks in various researches was mediocre and did not attract much attention. Then, with the development of hardware and the decrease of computing cost, deep learning once again appeared in people’s field of vision and made breakthroughs in various fields [6]. During the development of neural network for many years, several representative models have appeared. The Restricted Boltzmann Machine (RBM) is an extension of the Boltzmann Machine [7]. Since the connections between the same layers are removed, the learning efficiency is greatly improved. Autoencoder (AE) can be regarded as a variant of the traditional multilayer perceptron. In addition, there are many variants of autoencoders that make autoencoders better in text recommendation, rating prediction, and other issues. Modeling sequential data with a recurrent neural network (RNN) is possible [8]. When an RNN receives a new input, it combines it with the implicit state vector to produce an output that is dependent on the entire sequence. The RNN model includes various modifications to adapt to diverse needs and applications, such as long- and short-term memory (LSTM), gated recurrent unit (GRU), and so on. It is the first choice for natural language processing projects such as text and audio. Convolutional neural network (CNN) excels when dealing with data with grid-like topology. Compared to traditional multilayer perceptrons, CNNs employ a pooling technique to reduce the number of neurons in the model and improve shift or spatial invariance. Furthermore, CNN’s shared weight design can reduce the amount of parameters in the model, as well as its complexity. In recent years, CNN have become the first choice in the field of image processing [9].

Today, there are many breakthroughs in applying deep learning to recommender systems [10]. For example, in the recommendation algorithm research based on association rule recommendation combined with neural network [11], the recommendation algorithm research combining neural graph and collaborative filtering [12], and the recommendation algorithm research that applies deep learning to item-based collaborative filtering [13], such cross-domain research is a hot research topic today. According to the combination, there are two types of models: ensemble models and neural network models. Consider whether standard recommendation models should be combined with deep learning or used separately. There are several types of ensemble

models: ensemble models that integrate deep learning with traditional models, but only rely on deep learning models for recommendation. Also, attempt to blend deep learning techniques with classic recommendation approaches in a novel manner, and apply deep learning techniques to ensemble models in traditional recommendation techniques. According to the specific needs of the recommender system, different types of deep learning techniques and different combination methods will be used in order to obtain a better recommendation effect. When it comes to content recommendation systems, deep learning may be used to propose more contents for goods and users. High-level characteristics may be learned via deep learning using auxiliary data, capture the complex relationship of the data itself in rich information sources, and also obtain the difficult-to-obtain nonlinear relationship between users and recommended items. Based on deep learning in collaborative filtering recommendation systems, traditional collaborative filtering-based recommendations use user-item matrices. In practical applications, the rating matrix is usually very sparse, which leads to a decrease in recommendation performance. Some researchers take advantage of the feature that deep learning can learn effective representations and propose a neural network-based collaborative filtering method. A recommendation method based on autoencoder proposed by some scholars [14]. There is also the Collaborative Denoising Autoencoder (CDAE) [15], which is mainly used for ranking prediction. A restricted recommender system based on restricted Boltzmann machines has been proposed [16]. The researchers used a conditioned Boltzmann machine to incorporate implicit feedback information. There is also collaborative filtering based on recurrent neural networks. The basic idea is to use recurrent neural networks to model the impact of user historical sequence behaviors on users' current behaviors and then recommend items for users and predict user behaviors [17].

The basic idea of the hybrid recommendation method based on deep learning is to combine multiple recommendation methods. For example, some scholars proposed a hybrid model based on stacked denoising autoencoders [18]. The model exploits the latent association of users and items. The relationships in the auxiliary information are first learned using deep learning, and then the rating matrix is used for collaborative filtering. There is also the application of deep learning technology in social recommendation, and the application of deep learning in cross-domain recommendation and the combination methods used are also different [19]. For the recommendation in the field of constitution and constitutionalism, there is no open and perfect product in the industry. Most of the information products in the field of constitution and constitutionalism on the market are mainly based on document generation and error correction, and there is no product with case recommendation as the core function. In the academic field, lawyers' recommendation is the main research direction. There is only a small amount of research on the recommendation of case texts, and most of them are recommended by tags. In the face of unstructured texts, this kind of recommendation is less intelligent and effective [20, 21].

### 3. Method

*3.1. Basics of Deep Neural Networks.* Deep neural networks are the foundation of deep learning. The mainstream network structures DNN, CNN, and RNN in deep learning are all derived from the basic neural network. For decades, the research on neural networks has not been smooth sailing and has encountered several ups and downs, from the earliest perceptrons with only one layer of neural networks to two-layer neural networks with a single hidden layer, and then later a multilayer neural network containing several hidden layers, that is a deep neural network.

*3.1.1. Artificial Neural Network Basics.* To carry out deep learning research, you must first understand artificial neural network technology. The following introduces the neuron, neural network, forward and backward propagation, and simple parameter optimization method in artificial neural network. The smallest constituent element in a neural network is a neuron, which is also called a perceptron. The perceptron algorithm successfully solves many problems. Figure 1 illustrates a perceptron.

A perceptron has the following components, input weights: a perceptron can receive multiple inputs  $x_1, x_2, \dots, x_n$ . Each input contains a weight  $w_j$ , in addition to a separate bias term (bias)  $b$ , which is  $w_0$  in the above figure. In the activation function, there are many forms of activation function. For example, the step function  $f$  represented by the formula can be used as the activation function of the perceptron:

$$f(z) = \max(z, 0). \quad (1)$$

The output of the perceptron is calculated by the following formula:

$$y = f(\omega x + b). \quad (2)$$

In a neural network, a set of rules is used to link neurons. Figure 2 depicts the fundamental structure of a neural network. There are circular nodes for each neuron, and the arrows indicate connections between them. Obviously, the neural network in the figure is divided into four layers, the neurons in different layers are connected to each other, but the neurons in the same layer are not connected to each other. Secondly, the first layer on the left in the neural network above is mainly used to receive data, also known as the input layer; the first layer on the right is mainly responsible for the output of the neural network data, also known as the output layer; layering between the input layer and the output layer the term "hidden layer" refers to any layer that is not visible to the outer world. Compared with traditional neural networks, deep neural networks are generally defined as neural networks with more than 2 hidden layers. So the method of machine learning by using deep neural network is deep learning.

*3.1.2. Convolutional Neural Network Basics.* Convolutional neural network (CNN) also products derived from traditional neural networks. They are all composed of a series of neurons, but the difference from traditional neural networks is that convolutional neural networks tend to focus

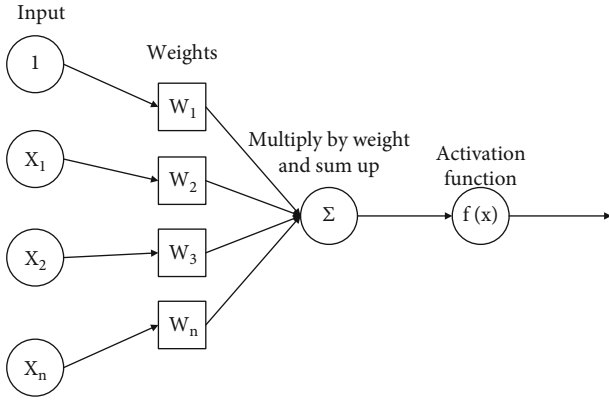


FIGURE 1: Perceptron algorithm.

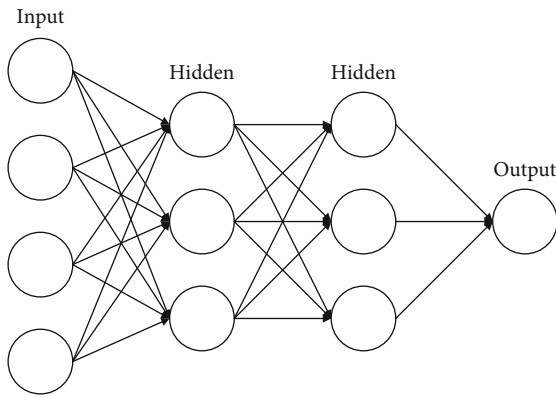


FIGURE 2: The basic structure of the neural network.

on input. These types of methods are often used in the field of computer graphics. Convolutional neural networks are characterized by the fact that neurons in a certain layer are connected to some neurons in the previous layer. This part of the region generates new neurons through convolution and nonlinear transformation. The entire convolutional neural network can be reflected with a differentiable end-to-end function, such as  $f : x \rightarrow y$ , where  $x$  is the original image and  $y$  is the class score. Convolutional neural networks generally use loss functions when adjusting and configuring network parameters. Compared with the traditional neural network, the convolutional neural network needs to set fewer parameters. It has the potential to enhance neural network training efficiency by adding more layers. The convolutional layer, the pooling layer, and the fully connected layer make up the majority of the network layer in the convolutional neural network. These three layers of the network are combined in many ways to create the fundamental structure of a convolutional neural network. CNN reduces the number of weights through the feature fitting ability in the convolutional layer, which can effectively reduce the computational burden. In the pooling layer, through feature compression, it can effectively retain the key structural information and reduce the amount of computation. Therefore, it is often used in image recognition and other fields.

**3.2. Semantic Similarity Calculation.** Text similarity calculation is the degree of semantic similarity of texts. Search engines, recommender systems, machine translation, and other tasks all rely more or less on similarity calculation as the basis. It can be seen how important it is in the field of NLP.

**3.2.1. Common Similarity Calculation Methods.** For  $N$ -gram similarity, based on the  $N$ -gram model, it is a fuzzy matching method that calculates the similarity by obtaining the difference between two sentences. Its calculation formula is shown below [22].

$$\text{Similarity} = |G_N(S)| + |G_N(T) - 2 * |G_N(S) \cap G_N(T)||, \quad (3)$$

where  $G_N(S)$  and  $G_N(T)$  represent the set of  $N$ -Grams in strings  $S$  and  $T$ , respectively, and  $N$  generally takes 2 or 3. The closer the strings are, the more similar they are, and when two strings are exactly equal, the distance is 0.

The computation of Jaccard similarity is rather straightforward. The ratio of the intersection and union of word sets in two sentences needs to be determined. Its calculation formula is as follows [23]:

$$S(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}, \quad (4)$$

where  $0 \leq J(A, B) \leq 1$ ,  $Jaccard(A, B) = 1$  when both  $A$  and  $B$  are empty.

This method is similar to the  $N$ -gram method in that the similarity is judged by the size of the same part of the sentence.

**3.2.2. Semantic Representation.** We need suitable semantic representations to represent the meaning of language reimplications in a suitable way for input to computers. There are many ways to express semantics. In the early days, the representation methods of topic models including One-hot, SVD, LDA, and other methods were proposed. One-hot representation helps to convert categorical values into binary vectors wherein the categorical values are firstly mapped to integer variables and then the integer value is represented as binary vector which consists of all zero values except the index of the integer marked as one. The Singular Value Decomposition (SVD) in ML is a matrix factorization technique which decomposes a matrix into three generic and familiar matrices and is used as a data reduction technique. The linear discriminant analysis (LDA) in ML uses Bayes' theorem to estimate probabilities and is used for dimensionality reduction. The features of higher dimensionality space are projected onto lower-dimensionality space to avoid dimensionality issues thereby reducing resource and dimensional costs. This kind of method has serious data sparse phenomenon, which leads to many problems. For example, the dimension of the vector obtained by using the One-hot method is the same as the length of the dictionary, resulting in a very high vector dimension and a huge amount of computation, and the vector itself is very sparse, wasting space, and computing power. Later, Wang et al. proposed a three-

layer neural network language model that globally shares the input layer parameters, which is the distributed representation of words [24].

**3.3. Court Information Text Classification Model Based on Deep Learning.** After obtaining a large number of case-related texts, it will be difficult to process due to the huge number and variety of original texts. Since this paper deals with the case as the center, classifying the text according to the type of the case will facilitate the subsequent use of the case text. To this end, this paper proposes a character-level text classification model OCCNN (One-hot Character Convolutional Neural Network) based on convolutional neural network. This paper preprocesses a large number of court judgment documents collected. A vocabulary is generated from the training data, using the One-hot method for text representation, preserving character-level information. Then, features are automatically extracted using convolutional neural networks for better classification predictions. The main model of OCCNN is divided into four parts: text preprocessing, text representation, neural network model training, and text classification, as shown in Figure 3.

**3.3.1. Character-Level Text Representation.** Judgments can be divided into administrative, civil, compensation, criminal, and execution judgments. After modifying them into a unified format, simple preprocessing is performed to remove redundant symbols and unrecognizable characters. The existence of the validation set is to solve the problem that deep learning is prone to overfitting in the form of cross-validation. To input the text into the neural network model to extract features, it is necessary to convert the text into a form that the model can recognize. In order to simplify the steps and shorten the training time, this paper chooses the most simple and efficient method, the One-hot representation, and then the general One-hot notation is improved at the character level to avoid the problem that the vector is too large and consumes too much computing power due to the large vocabulary of the traditional One-hot representation, and the space waste caused by the feature matrix is too sparse.

**3.3.2. Feature Extraction and Classification.** Convolutional networks can be used to extract information from the original signal, eliminating the need to manually design features and effectively reduce labor consumption. In this paper, character-level text information is input into a convolutional neural network, features are extracted, and a classification model is constructed. In character-level text representation, perform text representation according to the previous content, and generate a character-level text representation matrix. In a convolutional layer, after the text representation is over, enter the convolutional layer. As shown in the following formula, this step mainly does the convolution operation. First, take the training set of character-level text representations as the input matrix  $W_{k+1 \times N}$ . Then, set several parameters of the convolution kernel  $C^{1,a}$ , where 1 represents the height of the convolution kernel and  $a$  represents the vector dimension. Then, put the convolution kernel into

the input matrix and slide it according to the set step size, multiply the corresponding elements and sum them up, and the final result is the feature map obtained by convolution. In this step, several feature maps consistent with the number of convolution kernels are obtained, which are denoted as  $h^{1,a}$ , where  $h^{1,a}$  is the input matrix and the convolution kernel for convolution calculation and then add the operational sum of the bias  $b'$ .

$$h^{1,a} = f(x * C^{1,a} + b'). \quad (5)$$

The weight parameters need to be reduced through a pooling layer, also called a downsampling layer. The pooling operation reduces the computational complexity of the next fully connected layer while retaining important features. And it can effectively reduce overfitting and improve the fault tolerance of the model. The method used in this step is max-pooling, and the value after the maximum pooling operation is output after the pooling calculation. The feature map corresponding to each convolution kernel is operated, and then, the obtained maximum values are spliced together, and finally, a new feature vector representing the sentence is obtained. Then, the new feature vectors obtained by pooling are combined, and their categories are marked to obtain a feature pair in the form of  $\{(x_m, y_m)\}$ , where  $x_m$  represents the new feature vector obtained after pooling and  $y_m$  represents the text category. In a fully connected layer, dropout operation is performed after the fully connected layer and then connected to ReLU activation. Dropout operation can effectively improve overfitting when training the network. In the classification part, this paper chooses to classify by softmax function. Input the result of the fully connected network into the softmax function, map it to a value in the interval (0, 1); these values add up to get 1, and use it as the probability of being this class. Finally, this paper selects the class with the highest probability as the final classification result. The calculation formula used is as follows:

$$f(x)_\emptyset = \frac{1}{1 + \exp(-\emptyset^T x)}, \quad (6)$$

where  $\exp$  represents the exponential function with base  $e$ .

Here, the gradient descent method is selected to update the gradient, and the following formula is used as the loss function. The value of the loss function is calculated by the minimum cost function:

$$J(\emptyset) = \sum_{i=1}^M y(i) \log f_\emptyset(x^i). \quad (7)$$

The softmax function returns the probability of each output category and finally divides the classification category of the text according to the probability.

**3.3.3. Various Parameter Settings.** The parameters of the character-level convolutional neural network we use for training are set as follows: the size of the convolution kernel is 5, the number of convolution kernels is 128, and the

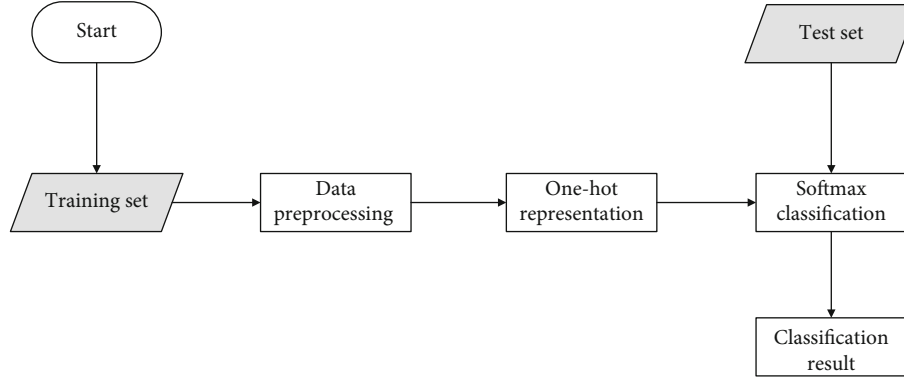


FIGURE 3: OCCNN-based text classification model.

TABLE 1: Convolutional neural network parameter setting.

Parameter meaning	Parameter values
Number of categories	5
The number of convolution kernels	256
Convolution kernel size	5
Fully connected layer neurons	128
Learning rate	0.01

activation function selects ReLU activation. The following formula is the ReLU expression:

$$\text{ReLU} = \begin{cases} x, & x > 0, \\ 0, & x \leq 0. \end{cases} \quad (8)$$

Since the result of ReLU in the nonnegative interval is constant, there is no gradient disappearance problem, so the model can converge stably. The ReLU function is simple to operate, making the training converge faster. Although the validation set is set to avoid overfitting through cross-validation, overfitting is still easy to occur during the training process because the training set is not large enough. Therefore, dropout is set after the fully connected layer to effectively reduce overfitting. This method can reduce the dependence of a neuron on some specific neurons and improve the robustness of the neural network model. This sets the dropout retention ratio of the model to 0.5. This paper chooses to set the learning rate to 0.01, and the experiment has basically converged after three iterations. After five rounds of training, the accuracy has no longer improved significantly. Other neural network parameter settings used in the experiment are shown in Table 1.

**3.3.4. Evaluation Standard Setting.** To verify the effectiveness of the OCCNN model, this paper performs classification on multiple text classification models. The text classification model of traditional machine learning algorithms requires labels, and it is too laborious to manually label a large number of texts, and the quality and number of labels have a huge impact on the classification results, so it is difficult to use machine learning algorithm-based classification models for comparative experiments. Finally, the baseline methods

used in the experimental comparison are two RNN improved models: LSTM model and GRU model. LSTM overcomes the problem that RNN cannot handle long-range dependencies well, and GRU is one of many variants of LSTM. Forget, input, and output gates are all implemented using LSTM. Update and reset gates are implemented in the GRU model. LSTM comes in a variety of flavors, but GRU keeps the power of LSTM while streamlining the implementation.

For the experimental results, it is necessary to select appropriate criteria to quantitatively judge whether the classification is successful. Here, the accuracy rate ( $P$ ), recall rate ( $R$ ), and comprehensive evaluation index ( $F1$ ) are selected as the criteria. As a standard, the accuracy rate has been able to effectively evaluate whether the classification result of a classifier is successful. However, in order to assess the classification effect from multiple aspects, the effect is more accurate. This paper first defines four classification cases, TP is judged to be correct, FP not this type is judged to be this classification, FN is judged to be other types, and TN nonclass is judged to be not this type. The formulas for precision, recall, and F1 value are as follows:

$$P = \frac{TP}{TP + FP}, \quad (9)$$

$$R = \frac{TP}{TP + FN}, \quad (10)$$

$$F1 = \frac{2PR}{P + R}. \quad (11)$$

**3.4. Semantic Similarity Calculation Model Based on Word Vector.** In the constitutional case recommendation system, we analyze user preferences through user input, and the items to be recommended are a large number of case documents and other documents. Since user interests and items to be recommended are both in text form. The core of a recommender system can be transformed into a similarity calculation problem. In order to enhance the in-depth understanding of the text, this paper uses a model combining keyword frequency and word vector to solve the similarity calculation problem. First, the Word2Vec model is trained on the preprocessed text in order to retrieve its vector corresponding to each preprocessed

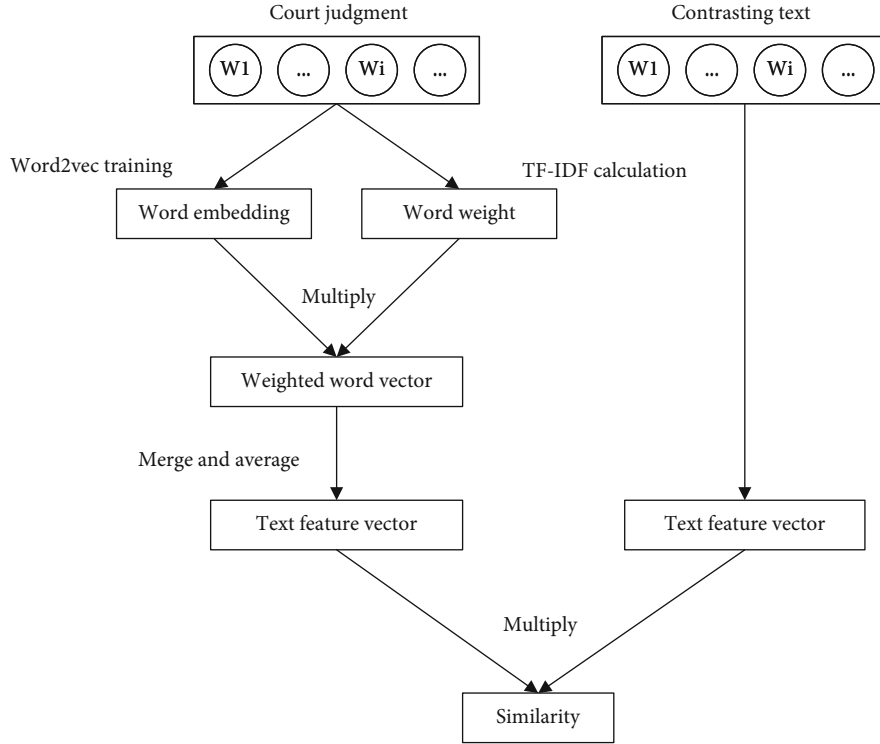


FIGURE 4: The flow chart of the acquaintance model.

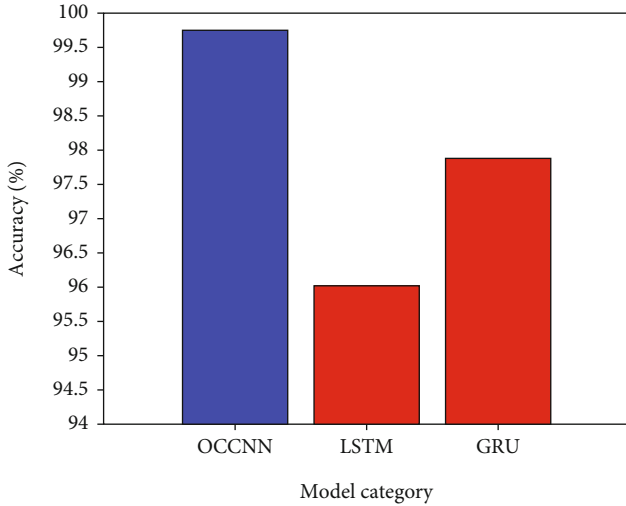


FIGURE 5: Classification accuracy of different models.

word. Use TF-IDF to get the weights of each word in the text. Using the word weight and word vector, the weighted word vector for each word is computed. The feature vector of the text is the mean value of the dimension of the feature matrix, which is created by concatenating the word vectors containing words from the text. The last step is to compute the cosine similarity in order to determine the degree of similarity. Figure 4 depicts the steps involved in the computation. Semantic similarity calculator is an automated approach for measuring similarity and its relatedness provides the necessary information for

semantic context information. This is extremely important for information retrieval related applications and other natural language processing tasks involving word sense disambiguation.

**3.4.1. Word Vector Training and Weight Calculation.** The paper uses the CBOW computational model. In order to avoid the phenomenon of gradient explosion in the calculation, choose negative sampling for calculation. To count the total amount of text material knowledge and the required accuracy, in order to avoid filtering out too many rare terms since the quantity of court judgment text is so little, this study sets the min-count in the Word2vec model to 2. Set the hidden layer of the neural network's dimension to 100 for the output word vector.

It is a popular statistical technique used in text mining and information retrieval. TF-IDF is an easy concept to grasp. As a general rule, the amount of significance that each given word has in a text is inversely related to the number of times it occurs in that document's corpus. TF stands for term frequency and is calculated as shown in the formula.  $f_{ij}$  represent the word frequency that appears in the  $j$ th document for the word  $t$ . The numerator  $n$  denotes the total number of occurrences of the term  $t$  in the  $j$ th document, as seen in the formula. The denominator is the total amount of words in the file that appear in the denominator.

$$tf_{ij} = \frac{n_{i,j}}{\sum_k n_{k,j}}. \quad (12)$$

IDF refers to Inverse Document Frequency. There are IDI and  $t_i$  in the denominator, which reflects the total

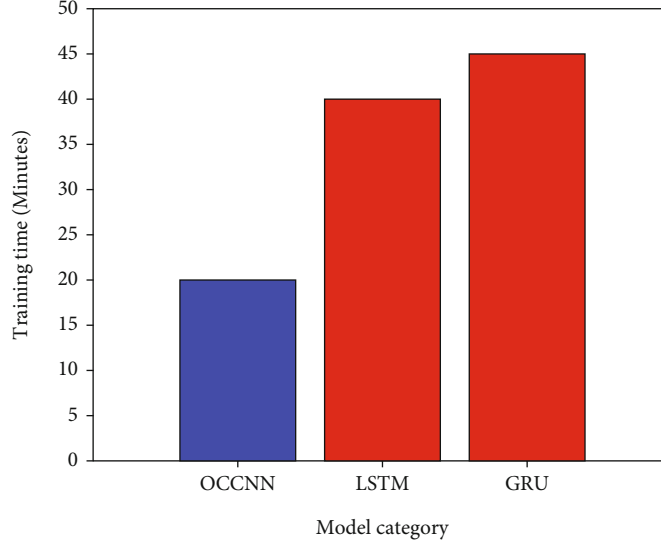


FIGURE 6: Comparison of training time of different models.

TABLE 2: Number and sources of judgments for each type of case.

Data source	A1	A2	A3	A4	A5	Total
A court	711	10256	1202	1523	1856	15548
Online case text	165	155	190	200	143	853

number of documents. The denominator will be zero if the text does not include the term  $t_i$ , and a calculation error will occur. Therefore, for the convenience of calculation, 1 is usually added to the denominator to avoid calculation problems. The calculation formula is as follows:

$$idf_i = \log \frac{|D|}{1 + |\{j : t_i \in d_j\}|}. \quad (13)$$

The calculation method of TF-IDF is the product of word frequency and inverse word frequency. In this way, the influence of high-frequency words on the calculation result can be ignored, and the real weight of keywords that are more important to a document can be truly obtained.

**3.4.2. Similarity Calculation.** Using the distance between vectors, the model estimates how similar the text is to each other. Common distance measures include Euclidean distance and cosine similarity. However, since the value range of  $\cos \theta$  is -1 to 1, in practice, this paper prefers to normalize it between 0 and 1, So the formula reveals the true way for determining cosine similarity:

$$\text{Similarity} = 0.5 + 0.5 * \cos \theta = 0.5 + 0.5 * \frac{X \cdot Y}{\|X\| \|Y\|}. \quad (14)$$

Since cosine similarity is compared with Euclidean similarity and other calculation methods, the angle of the vector, not the length of the vector, determines the computation

outcome. So it is widely used in the calculation of semantic similarity, and this paper also chooses to use the algorithm.

**3.4.3. Evaluation Criteria Setting.** In order to verify the effectiveness of the TF-W2V model, this paper uses a total of 18,368 judicial texts as the data source and uses different types of similarity calculation methods for comparative experiments, including the TF-IDF algorithm that performs the best in the traditional model, the computationally fast Simhash algorithm, and the deeply semantically understood Doc2Vec algorithm. The Simhash algorithm is extremely efficient in identifying duplicates in web documents. It uses a fingerprint technique with features that fingerprints of duplicates differ only in case of smaller number of bit positions. The Simhash fingerprint is generated for each of the objects, and when two objects have similar fingerprints, they are considered as near-duplicates. The Doc2vec algorithm helps in generating numerical representation of a document regardless of its length. The algorithm uses a paragraph vector which is unique to each paragraph. The paragraph vectors are requested to predict the next word considering data pertinent to contexts sampled from paragraph [25]. The efficiency of the TF-W2V model is shown by comparing its experimental results to those of various other algorithm models. There are two experiments conducted in this paper: an accuracy experiment and an experiment based on DCG (Distributed Computational Geometry). The accuracy rate (precision) is utilized as an assessment metric in the accuracy rate experiment, and the method for calculating the accuracy rate is as follows. For assessment, we utilize the text's original label. This study will consider the experimental findings positive if they yield the same sorts of instances; if they yield different types of cases, it will consider them negative; this paper will judge it as negative, that is, not similar.

$$\text{Precision} = \frac{tp}{tp + fp} = \frac{tp}{n}. \quad (15)$$



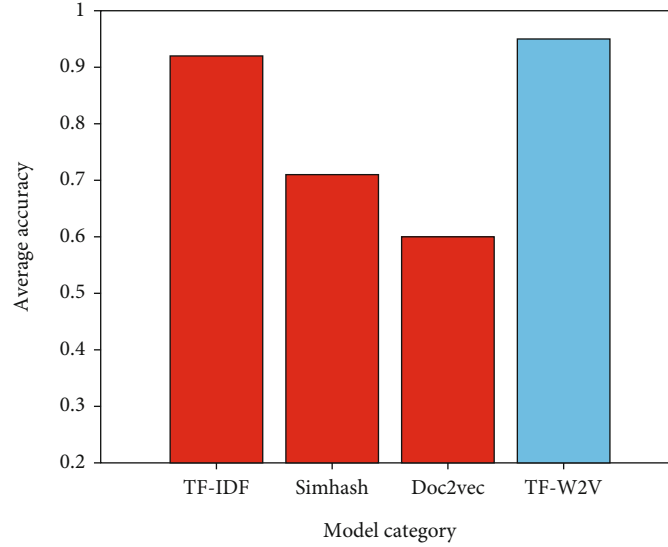


FIGURE 7: Bar comparison chart of average accuracy experimental results.

Content-based recommendation technology originated from information retrieval and information filtering. Using merely the model's accuracy to assess the model's accuracy is a challenging task. The DCG (Discounted Cumulative Gain) assessment index, which is extensively employed by search engines, is utilized in this work. When employing the DCG technique for evaluation, the following assumptions must be made. High correlation results are listed first in the list of results that were returned. The correlation between high and low scores in rank confirmation is stronger for high scores. The first step is to score the outcomes of the computation. There are four classifications here: strong resemblance, moderate similarity, low similarity, and total dissimilarity. Three points, two points, one point, and zero points are the equivalent scores. Formulas for DCG's computation are presented below.

$$DCG_p = \sum_{i=1}^p \frac{rel_i}{\log(i+1)}, \quad (16)$$

where  $DCG_p$  represents the DCG value of the first  $P$  results retrieved and  $rel_i$  represents the relevant grade score of the  $i$ -th retrieval result.

## 4. Experiment and Analysis

**4.1. OCCNN Model Experimental Results.** The experimental data uses court judgments in five categories of administrative, civil, compensation, criminal, and enforcement as the data source, and the data comes from the real data of a certain court. Each category has 2600 texts, for a total of 13000 texts. Use the judgment category as a single label. After classification, the text training set, test set, and validation set are separately integrated and labeled and preprocessed to remove redundant symbols and unrecognizable parts, then unified format conversion. In this paper, the sequence length is selected as 1000, and the average length of the training set

text is about 2000. However, since many words in the original text are not in the dictionary, at the end of the text, there are a lot of information that is invalid for classification, such as time, place, judges, and the fixed words of the judgment. Finally, after many trials, this paper chooses 1000 as the optimal sequence length.

Calculate the average classification accuracy of the three models on each type of material. And calculate the average of the elapsed time for several trials. A histogram of the average classification accuracy and training time of each model is made, as shown in Figures 5 and 6.

As can be seen from Figures 5 and 6, the average accuracy of each neural network is 99.75%, 96.02%, and 97.88%, respectively. Compared with the LSTM model, the accuracy of the OCCNN model is increased by 3.73 percentage points, and compared with the GRU model by 1.87 percentage points, the effect is the best. The other two models' categorization accuracy is likewise high. However, the OCCNN model has a better training time. It takes longer to train the GRU model than the OCCNN model or the LSTM model.

**4.2. Experimental Results of the TF-W2V Model.** In terms of experimental data, this paper uses five categories of court judgments, including administrative cases A1, civil cases A2, compensation cases A3, criminal cases A4, and enforcement cases A5 as data sources. The obtained case texts were 853, and the texts were classified by the text classification method. Table 2 shows the number and sources of judgments for each type of case.

The 16401 pieces of data were calculated by TF-IDF algorithm, Simhash algorithm, Doc2Vec algorithm, and TF-W2V model, respectively, and the results were calculated by the accuracy rate and DCG@10 evaluation index to obtain the average value. Figures 7 and 8 show a schematic diagram of the comparison of the average results obtained by testing the accuracy and DCG (@10) as the evaluation index.

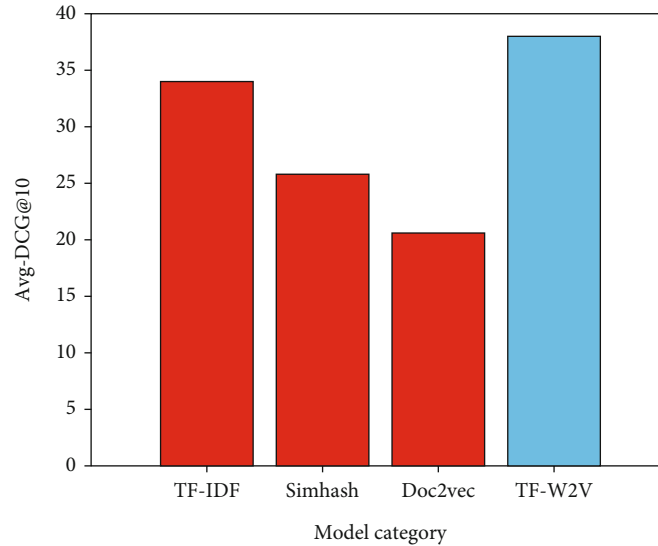


FIGURE 8: Bar comparison chart of average DCG@10 experimental results.

It can be seen from the average results of the accuracy experiment that compared with the baseline method, the TF-W2V used in this paper has the best results, but it is only a little better than the TF-IDF algorithm, and the accuracy of the fastest Simhash algorithm is second. The relative results of Doc2vec modified from Word2Vec are slightly worse. From the average results of the DCG@10 experiment, the TF-W2V similarity calculation effect used in the paper is much better than TF-IDF, followed by the Simhash model, and the Doc2vec model has poor results. It can be seen that compared with other single calculation models, the TF-W2V model combined with word frequency and word vectors used in this paper is closer to the effect of expert evaluation, and the comprehensive evaluation is better than the results obtained by other algorithms.

## 5. Conclusion

This paper proposes OCCNN court judgment classification method based on convolutional neural network. Collect a large amount of text information of court judgments, and use word embedding method for text representation, which quickly and efficiently realizes the mapping of text to vectors, saving the time of training word vectors. The resulting matrix is then fed into the convolutional neural network for training. The local connection and weight sharing technology of convolutional neural network can effectively extract feature information, and only a small amount of training time was used. The experimental results show that the model can achieve a classification accuracy of 99.75% on the test set, and the training time is only 50% of the commonly used recurrent neural network algorithm. Using this model, the collected case texts can be effectively classified, which is convenient for subsequent use of the case texts. This paper proposes a similarity calculation model TF-W2V. After preprocessing the text, TF-W2V trains the word vector through Word2Vec and obtains the weight of each word

through the TF-IDF algorithm. Then, the weighted average of the word vector and weight of each text is obtained, and the text matrix is jointly obtained. After the compression matrix is averaged, the eigenvectors of the text are obtained. Finally, through the cosine similarity algorithm, the similarity of the documents is obtained. After the model compares the target text with the compared text set, it returns similar documents and the similarity between the target document and the document in descending order of similarity. It can be seen from the experiments that the calculation speed of this algorithm is stable, and compared with other traditional similarity calculation models, the highest accuracy is greatly improved. The model is evaluated on the basis of training time and accuracy which justify its effectiveness in comparison to the other state of the art approaches. However, the metrics considered could be increased to build more confidence on the applicability of the approach. Hence, as part of future research metrics namely precision, recall, sensitivity, and specificity could be included to strengthen the justification of its superiority.

## Data Availability

The datasets used during the current study are available from the corresponding author on reasonable request.

## Conflicts of Interest

The authors declare that they have no conflict of interest.

## Acknowledgments

This work was supported by the support of the first-class teaching team of Qiannan Normal University for Nationalities (Project No.: 2021xjg013).

## References

- [1] D. H. Mcknight, V. Choudhury, and C. Kacmar, "Developing and validating trust measures for e-commerce: an integrative typology," *Information Systems Research*, vol. 13, no. 3, pp. 334–359, 2002.
- [2] S. Zhang, B. Verma, D. Tjondronegoro, and V. Chandran, "Facial expression analysis under partial occlusion," *Computing surveys*, vol. 51, no. 2, pp. 1–49, 2019.
- [3] J. G. Liu, T. Zhou, B. H. Wang, and Y. C. Zhang, "Highly accurate recommendation algorithm based on high-order similarities," *Physics*, vol. 389, 2008.
- [4] J. Pei, K. Zhong, M. A. Jan, and J. Li, "Personalized federated learning framework for network traffic anomaly detection," *Computer Networks*, vol. 209, article 108906, 2022.
- [5] R. Mu, "A survey of recommender systems based on deep learning," *IEEE Access*, vol. 6, pp. 69009–69022, 2018.
- [6] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [7] H. Chen and A. F. Murray, "Continuous restricted Boltzmann machine with an implementable training algorithm," *IEEE Proceedings-Vision Image and Signal Processing*, vol. 150, no. 3, pp. 153–158, 2003.
- [8] H. Y. Lee, E. E. Giorgi, B. F. Keele et al., "Modeling sequence evolution in acute HIV-1 infection," *Journal of Theoretical Biology*, vol. 261, no. 2, pp. 341–360, 2009.
- [9] W. Duan, J. Gu, M. Wen, G. Zhang, Y. Ji, and S. Mumtaz, "Emerging technologies for 5G-IoV networks: applications, trends and opportunities," *IEEE Network*, vol. 34, no. 5, pp. 283–289, 2020.
- [10] H. Tong, C. Zhang, and J. Hu, "Design and research of mobile learning recommendation system based on deep learning perspective," *Journal of Physics: Conference Series*, vol. 1915, no. 2, article 022065, 2021.
- [11] M. Karabatak and M. C. Ince, "An expert system for detection of breast cancer based on association rules and neural network," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3465–3469, 2009.
- [12] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 5–53, 2004.
- [13] F. Xue, X. He, X. Wang, J. Xu, K. Liu, and R. Hong, "Deep item-based collaborative filtering for top-n recommendation," *ACM Transactions on Information Systems (TOIS)*, vol. 37, no. 3, pp. 1–25, 2019.
- [14] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P. A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, no. 12, pp. 3371–3408, 2010.
- [15] H. Lei and Y. Yang, "CDAE: a cascade of denoising autoencoders for noise reduction in the clustering of single-particle Cryo-EM images," *Frontiers in Genetics*, vol. 11, 2021.
- [16] M. Fu, H. Qu, Z. Yi, L. Lu, and Y. Liu, "A novel deep learning-based collaborative filtering model for recommendation system," *IEEE Transactions on Cybernetics*, vol. 49, no. 3, pp. 1084–1096, 2019.
- [17] C. Wu, J. Wang, J. Liu, and W. Liu, "Recurrent neural network based recommendation for time heterogeneous feedback," *Knowledge-Based Systems*, vol. 109, pp. 90–103, 2016.
- [18] S. Sasaki, "Cross-talk noise reduction technique by shifted-stacked pair-lines structure," *Journal of Japan Institute of Electronics Packaging*, vol. 7, no. 6, pp. 548–555, 2004.
- [19] V. Rajyalakshmi and K. Lakshmana, "A review on smart city - IoT and deep learning algorithms, challenges," *International journal of engineering systems modelling and simulation*, vol. 13, no. 1, pp. 3–26, 2022.
- [20] G. Cai, Y. Fang, J. Wen, S. Mumtaz, Y. Song, and V. Frasca, "Multi-carrier M-ary DCSK system with code index modulation: an efficient solution for chaotic communications," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 6, pp. 1375–1386, 2019.
- [21] W. Naptali, M. Tsuchiya, and S. Nakagawa, "Class-based N-gram language model for new words using out-of-vocabulary to in-vocabulary similarity," *IEICE Transactions on Information and Systems*, vol. E95.D, no. 9, pp. 2308–2317, 2012.
- [22] R. Rousseau, "Jaccard similarity leads to the Marczewski-Steinhaus topology for information retrieval," *Information Processing & Management*, vol. 34, no. 1, pp. 87–94, 1998.
- [23] A. Ishai, L. G. Ungerleider, A. Martin, J. L. Schouten, and J. V. Haxby, "Distributed representation of objects in the human ventral visual pathway," *Proceedings of the National Academy of Sciences*, vol. 96, no. 16, pp. 9379–9384, 1999.
- [24] Y. Wang, D. Zhu, B. Zhang, Q. Guo, F. Wan, and N. Ma, "Review of data scraping and data mining research," *Journal of Physics: Conference Series*, vol. 1982, no. 1, article 012161, 2021.
- [25] M. E. Kumar, G. T. Reddy, K. Sudheer et al., "Vehicle theft identification and intimation using gsm & iot," *IOP Conference Series: Materials Science and Engineering*, vol. 263, no. 4, article 042062, 2017.