TdBrnn: An Approach to Learning Users' Intention to Legal Consultation with Normalized Tensor Decomposition and Bi-LSTM

Xiaoding Guo[1], Hongli Zhang[1,*], Lin Ye[1] and Shang Li[1]

Abstract: With the development of Internet technology and the enhancement of people's concept of the rule of law, online legal consultation has become an important means for the general public to conduct legal consultation. However, different people have different language expressions and legal professional backgrounds. This phenomenon may lead to the phenomenon of different descriptions of the same legal consultation. How to accurately understand the true intentions behind different users' legal consulting statements is an important issue that needs to be solved urgently in the field of legal consulting services. Traditional intent understanding algorithms rely heavily on the lexical and semantic information between the original data, and are not scalable, and often require taxing manual annotation work. This article proposes a new approach TdBrnn which is based on the normalized tensor decomposition method and Bi-LSTM to learn users' intention to legal consulting. First, we present the users' legal consulting statements as a tensor. And then we use the normalized tensor decomposition layer proposed by this article to extract the tensor elements and structural information of the original tensor which can best represent users' intention of legal consultation, namely the core tensor. The core tensor relies less on the lexical and semantic information of the original users' legal consulting statements data, it reduces the dimension of the original tensor, and greatly reduces the computational complexity of the subsequent Bi-LSTM algorithm. Furthermore, we use a large number of core tensors obtained by the tensor decomposition layer with users' legal consulting statements tensors as inputs to continuously train Bi-LSTM, and finally derive the users' legal consultation intention classification model which can comprehensively understand the user's legal consultation intention. Experiments show that our method has faster convergence speed and higher accuracy than traditional recurrent neural networks.

Keywords: Normalized tensor decomposition, Bi-LSTM, legal consultation, users' intention.

1 Introduction

With the development of internet technology and the public's increasing awareness of the rule of law, online legal consultation has become an important means for people to conduct legal consultation. So how to understand the true intentions of users' legal

[1] School of Computer Science and Technology, Harbin Institute of Technology, Harbin, 150001, China.

[*] Corresponding Author: Hongli Zhang. Email: zhanghongli@hit.edu.cn.

Received: 29 May 2019; Accepted: 01 July 2019.

consulting statements? The significant issue to solving the above problems is to classify the legal consulting statements of users who have different legal knowledge backgrounds and language descriptions.

However, the intention understanding of users for legal consulting services is rarely mentioned in previous research works. The traditional methods of intent classification rely heavily on the grammar and semantic information between the vocabulary of the original data, and often requires a large number of manual annotation work. At the same time, the obtained intent classification model is poorly scalable, and its accuracy is not high and highly volatile.

This article proposes an intention understanding method TdBrnn for users' legal consulting statements by using the normalized tensor decomposition method and Bi-LSTM. We use the structure of tensor to represent original users' legal consulting statements, and use the normalized tensor decomposition layer proposed by this article to extract the most representative tensor elements and tensor structure information from the original user's legal consulting statement tensor, that is the core tensor. Compared with the original tensor representing the user's legal consulting statement, the core tensor has weaker reliance on the grammar and semantic information of the vocabulary in the original users' legal consulting statements dataset, and it removes the redundant information in the original tensor that has no value to the classification model of users' legal consultation intention.

It is worth mentioning that the use of normalized tensor decomposition layer proposed by us reduces the dimension of the original tensor, and also greatly reduces the computational complexity of the subsequent Bi-LSTM classification algorithm. In fact, our tensor decomposition layer implements the main information extraction operation for the original tensor which represent the user's legal consulting statement. Then we input the set of core tensors on behalf of users' legal consulting statements into the Bi-LSTM to train the model, and finally derive the deep neural network model for the intent classification of intention in users' legal consulting statements. Experiments show that our method has faster convergence speed and higher accuracy than traditional recurrent neural networks, and has better scalability and stability. One of the main reasons is the implementation of the tensor decomposition layer.

The contribution of this article can be summarized as the following:

1) *In this article, the user's legal consulting statement is expressed by tensor, and the normalized tensor decomposition layer is used to extract the main tensor element and tensor structure information in the original tensor, that is the core tensor.* Compared with the original tensor, the core tensor represents the most valuable users' legal consulting information in the original tensor, and it relies weaker on the grammar and semantic information of the user's original legal consulting statement. The core tensor greatly reduces the dimensions of the original tensor. At the same time, the computational complexity of the users' legal consultation intention classification model of subsequent Bi-LSTM is greatly reduced.
2) *This article proposes a new approach to understanding the intent of users' legal consultation.* We combine the normalized tensor decomposition layer with the Bi-LSTM, and use the core tensors obtained by the normalized tensor decomposition layer

to train the Bi-LSTM, and then construct a classification model of users' legal consultation intention based on deep learning. Compared with the direct use of the original tensor, our method has weaker dependence on the vocabulary and grammar information of the original users' legal consultation data, and has faster convergence speed and higher accuracy. More importantly, our approach is more stable and scalable.

3) *We conduct multiple comparison experiments on the users' legal consultation intention classification model TdBrnn and traditional recurrent neural networks.* In order to prove the advanced nature of our proposed method, we compare the Bi-LSTM with tensor decomposition layer with the traditional recurrent neural networks to test the accuracy of classification model of users' legal consulting intention.

The sections in this article are distributed as follows. Section 2 introduces the research status of users' legal consulting services in recent years. Section 3 provides the background knowledge required for the Bi-LSTM with tensor decomposition layer proposed in this article. Section 4 explains in detail the algorithm principle of TdBrnn which is based on the Bi-LSTM with normalized tensor decomposition layer proposed in this article. Section 5 gives a comparison experiment between the proposed method and other classical recurrent neural networks.

2 Related work

At present, there are relatively few studies on the understanding of users' legal consulting intention. Research on the intersection of computers and law fields has focused on two main areas. That is, the classification of legal cases and the judgment of legal cases.

Galgani et al. [Galgani, Compton and Hoffmann (2015)] proposed a legal citation classification system. The system uses a background corpus and regular expressions to construct a framework for legal citation classification. Galgani and Compton created legal citation classification rules by extending the traditional rippdown-down Rules method. Capuano et al. [Capuano, Maio, Salerno et al. (2014)] proposed an automatic classification method based on natural language expression for legal cases. The approach relies on existing knowledge bases of legal ontology and legal common sense. The method consists of three parts; (1) Retrieve topics related to the given legal ontology term from Wikipedia knowledge base. (2) Extract relevant concepts from given legal cases text. (3) Match knowledge ontology terms with relevant legal concepts extracted.

Sulea et al. [Sulea, Zampieri, Malmasi et al. (2017)] explored applications of text classification methods in legal related business. They used machine learning algorithms to predict the outcome of cases in French Supreme Court and its accuracy. They proposed an approach to get information automatically. This method largely obstructs irrelevant information in the complete case description and captures key information that affects the final decision in legal cases. Previous studies mainly used traditional text classification algorithms to classify textual information in legal field [Kanapala, Pal and Pamula (2017)]. However, traditional text categorization methods rely heavily on lexical and grammatical information of legal text data. And it requires a lot of manual labeling work and professional legal knowledge background support which requires a lot of manpower and time.

The essence of the above research is the application of text classification methods in the legal field. In Zhang et al. [Zhang and Oles (2001)], Zhang et al. discussed the performance of support vector machines and linear classification algorithms in text classification, and constructed a regularized linear classification framework. In Chen et al. [Chen, Huang, Tian et al. (2009)], Chen et al. optimized original Bayesian strategy and proposed a new Bayesian algorithm for text feature extraction. Zhang et al. [Zhang, Yoshida and Tang (2008)] proposed a text classification algorithm based on multi-words and support vector machine, which combines the concept of multi-words representation with text classification algorithms. The traditional text classification focuses on the use of machine learning algorithms which have strong dependence on vocabulary and grammar information in the original data. Therefore, these algorithms tend to have high volatility in classification accuracy and poor scalability.

3 Preliminaries

This chapter is divided into four sections. Section 3.1 provides the related notions and definitions used in this article. Section 3.2 gives a mathematical description of the tensor decomposition. Section 3.3 details the forward propagation process of the recurrent neural network LSTM. Section 3.4 gives a formal description of the problem of the classification of intention in users' legal consulting statements to be solved in this article.

3.1 Notations and definitions

We use lowercase letters to represent scalars (a,b), bold lowercase letters for vectors ($\mathbf{a},\mathbf{b}$), uppercase letters for matrices (A,B), and Euler letters for tensors (χ,ν). We use A^T to indicate the transpose of matrix A.

Definition 3.1: (The Outer Product of Vectors). Given two vectors $\mathbf{a}$ and $\mathbf{b}$, $\mathbf{a}\in\mathbb{R}^P$ and $\mathbf{b}\in\mathbb{R}^Q$. The outer product of $\mathbf{a}$ and $\mathbf{b}$ is defined as $C=\mathbf{a}\circ\mathbf{b}$, $C\in\mathbb{R}^{P\times Q}$.

$C(p,q)=\mathbf{a}(p)\mathbf{b}(q)$

Definition 3.2: (The elongation or compression operation). Given a matrix C, $C\in\mathbb{R}^{P\times Q}$, and a vector $\mathbf{s}$, $\mathbf{s}\in\mathbb{R}^P$. The elongation and compression operation on matrix C is expressed as $Z=C\times_S\mathbf{s}$, $Z\in\mathbb{R}^{P\times Q}$.

$$C\times_S\mathbf{s}=\begin{bmatrix} C(1,1)\mathbf{s}(1) & C(1,2)\mathbf{s}(1) & \cdots & C(1,Q)\mathbf{s}(1) \\ C(2,1)\mathbf{s}(2) & C(2,2)\mathbf{s}(2) & \cdots & C(2,Q)\mathbf{s}(2) \\ \vdots & \vdots & \vdots & \vdots \\ C(P,1)\mathbf{s}(P) & C(P,2)\mathbf{s}(P) & \cdots & C(P,Q)\mathbf{s}(P) \end{bmatrix}$$

And so on, given an N-mode tensor χ, $\chi\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$, and a matrix U, $U\in\mathbb{R}^{I_2\times I_1}$. The elongation or compression operation on χ and U is expressed as $\varsigma=\chi\times_S U$, $\varsigma\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$.

$\varsigma(i_1,\cdots,i_n)=\chi(i_1,\cdots,i_n)U(i_2,i_1)$

Definition 3.3: (The Frobenius Norm of a Tensor). χ is an N -mode tensor, that $\chi \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. Then $\|\chi\|_F$ represents the Frobenius norm of χ .

$$\|\chi\|_F = \sqrt{\sum_{i_1}^{I_1} \cdots \sum_{i_N}^{I_N} \chi(i_1, \cdots, i_N)^2}$$

Definition 3.4: (The n -mode Matricization of a Tensor). Given an N -mode tensor ν , $\nu \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$. The n -mode matricization of ν is represented as ν_n , $\nu_n \in \mathbb{R}^{J_n \times J_1 \cdots J_{n-1} J_{n+1} \cdots J_N}$. ν_n is calculated by fixing the n -mode dimension and arranging the elements of other dimensions in series.

Definition 3.5: (The n -mode Product of a Tensor and a Matrix). ν is an N -mode tensor, $\nu \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$, and C is a matrix, $C \in \mathbb{R}^{J_n \times I}$. The n -mode product of ν and C is represented as $\kappa = \nu \times_n C$, $\kappa \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_{n-1} \times I \times J_{n+1} \times \cdots \times J_N}$.

$$\kappa(j_1, \cdots, j_{n-1}, i, j_{n+1}, \cdots, j_N) = \sum_{m=1}^{J_n} \chi(j_1, \cdots, j_{n-1}, m, j_{n+1}, \cdots, j_N) C(m, i)$$

Tab. 1 gives all the symbols and definitions used in this article.

Table 1: Table of symbols

Symbol	Definition
x	Scalar
$\mathbf{a}$	Vector
C	Matrix
ν	Tensor
C^T	Transpose of matrix C
$\nu(j_1, \cdots, j_N)$	The ($j_1, \cdots, j_N$)th entry of ν , same for vetors and matrices
$\circ$	The outer product
$\times_S$	The elongation or compression operation
$\|\nu\|_F$	The Frobenius norm of tensor \nu
$\times_n$	The n -mode product

3.2 Tensor decomposition

Tensor decomposition is a process of decomposing the original tensor into a core tensor and a series of factor matrices [Mcneice and Jones (2001)]. The mathematical description of the tensor decomposition [De Lathauwer (2009)] operation is as follows:

Given an N -mode tensor $\nu \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$, the tensor decomposition on ν can be expressed as

$$\nu \approx \gamma \times_1 C_1 \times_2 C_2 \times \cdots \times_N C_N$$

where γ is the core tensor of ν, $\gamma \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. $\{C_n\}$ is its corresponding factor matrix set, $C_n \in \mathbb{R}^{I_n \times J_n}$. γ and $\{C_n\}$ satisfy the condition that minimize the objective function ϕ, where

$$\phi = \left\| \nu - \gamma \prod_n^N \times_n C_n \right\|_F$$

3.3 LSTM

Recurrent neural networks perform well for processing time series data, but they face many problems when dealing with long sequence dependent information, such as the phenomenon of gradient disappearance or gradient explosion. The Long Short-Term Memory (LSTM) solves these problems by setting a long-term dependency state and a gated structure [Greff, Srivastava, Koutnik et al. (2015)].

LSTM sets a unit state c_t to store long-term unit status at time t. Simultaneously LSTM [Gers, Schmidhuber and Cummins (2000)] set up three gates, which are the forget gate f_t, the input gate i_t and the output gate o_t. The role of these three gates is given below.

1) The forget gate f_t is used to control how much of the long-term unit state c_{t-1} at the last moment $t-1$ is retained to the long-term unit state c_t at the current moment t.
2) The input gate i_t is used to control how much of the current unit state $\widetilde{c_t}$ at time t is retained into the current long-term unit state c_t.
3) The output gate o_t is used to control how much of the long-term unit state c_t at the current time is retained into the output unit state h_t at time t.

The forward propagation process formula of LSTM is as follows:

$$\begin{bmatrix} f_t \\ i_t \\ o_t \\ \widetilde{c_t} \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} \left(\begin{bmatrix} w_{fh}, w_{fx} \\ w_{ih}, w_{ix} \\ w_{oh}, w_{ox} \\ w_{ch}, w_{cx} \end{bmatrix} \cdot \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} + \begin{bmatrix} b_f \\ b_i \\ b_o \\ b_c \end{bmatrix} \right) \tag{1}$$

$$c_t = f_t * c_{t-1} + i_t * \widetilde{c_t} \tag{2}$$

$$h_t = o_t * \tanh(c_t) \tag{3}$$

where x_t is the input of the LSTM unit at time t, h_{t-1} is the output of the LSTM unit at time $t-1$. $\widetilde{c_t}$ is the unit state at time t. σ is the logistic sigmod function,

$\sigma(x)=\frac{1}{1+e^{-x}}$. \$tanh\$ is the hyperbolic tangent function, $tanh(x)=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$. w_{fh} is the weight of the forgot gate f_t for h_{t-1}, w_{fx} is the weight of f_t for x_t, and the same is true for w_{ih}, w_{ix}, w_{oh}, w_{ox}, w_{ch} and w_{cx} relative to the input gate i_t, the output gate o_t and the unit state $\widetilde{c_t}$ at time t. b_f, b_i, b_o and b_c are the bias terms for f_t, i_t, o_t and $\widetilde{c_t}$ respectively. $*$ is the operation of multiplying by elements among matrices.

3.4 Mathematical description of the problem

In this paper, we present a new approach to understanding the intention of users' legal consulting statements, namely TdBrnn. The intention here refers to the motivation of the users' legal consulting statements, including the case handling process, crimes of legal cases, basic knowledge of the law, the related punishment, etc.

TdBrnn represents user's legal consulting statements as tensors and decomposes them into core tensors using the normalized tensor decomposition method. TdBrnn uses the obtained core tensors as inputs to the subsequent Bi-LSTM, and finally completes the classification of users' legal consulting statements.

The formal description of users' legal consultation intention understanding problem to be solved in this article is as follows:

Problem 1: Given a set of users' legal consulting statements Z, $Z=\{(\nu^{(1)},L^{(1)}),(\nu^{(2)},L^{(2)}),\cdots,(\nu^{(N)},L^{(N)})\}$, where $\nu^{(n)}$ represents the user's legal consulting statement, $L^{(n)}$ represents the type of consultation of the user's legal consulting statement corresponding to $\nu^{(n)}$. Our goal is to train a Bi-LSTM classification model Ω for the classification of intention in users' legal consulting statements.

4 Our approach

In this chapter, Section 4.1 introduces the normalized tensor decomposition method proposed in this article. Section 4.2 details the classification algorithm TdBrnn for the intent understanding of users' legal consulting statements.

4.1 The normalized tensor decomposition method

4.1.1 Tensor representation of users' legal consulting statements

In this article, we present the user's legal consulting statement in three-dimensional tensor [Sidiropoulos, De Lathauwer, Fu et al. (2017)]. The first dimension represents modules in users' legal consulting statements. The second dimension represents the crucial feature elements in each module, and the third dimension represents word vectors of the above crucial element vocabulary.

This article divides the user's legal consulting statement into five modules, which are the subject module, the object module, the behavior module, the consequence module, and the special situation module. The subject and object modules represent the subject and object in the user's legal consulting statement, respectively. The behavior module refers to the important action information involved in the consultation event. The consequence module represents the final purpose of the consultation. The special situation module refers to special cases that are dedicated to legal field, such as underage, national level protected cultural relics, relief supplies, public places, etc.

For example, the user's legal consulting statement is known as "Xiao li stole $ 50,000 from the store in the absence of the owner, and refused to return the money. How should he be convicted and punished", the subject module includes "Xiao Li" and "owner", and the object module includes "$ 50,000", the behavior module includes "stole" and "refused to return", the consequence module includes "convicted and punished", and the special situation module includes "in the absence of the owner" and "store".

In order to represent the user's legal consultation statement in tensor, we perform the following operations on the original consultation statement data. (1) Module division; (2) Meaningless lexical filtering; (3) Word vector representation. As shown in Fig. 1, module division refers to dividing original users' legal consulting statements into the above five modules, namely the subject module, the object module, the behavior module, the consequence module, and the special situation module. Meaningless lexical filtering refers to removing meaningless words in original sentences, such as repetition, noise vocabulary, etc. Word vector representation refers to using word vectors to represent remaining words. Finally, original tensors are obtained and used as inputs of neural networks.

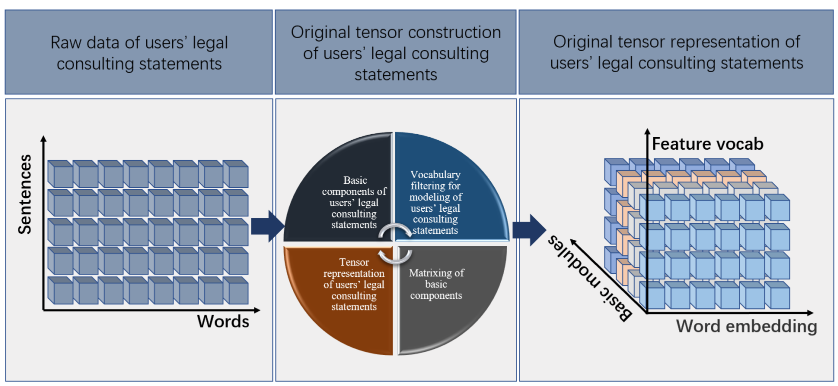

Figure 1: Tensor representation of users' legal consulting statements

4.1.2 The normalized tensor decomposition on users' legal consulting statements

In this article, we propose the normalized tensor decomposition method to decompose original tensors that represent users' legal consulting statements into core tensors. We represent the user's legal consulting statement in a three-mode tensor. The first mode represents the five modules included in the user's legal consulting statement, they are the subject module, the object module, the behavior module, the consequence module, and the situation module. The second mode represents the influential words contained in the corresponding module. The last mode is used to store the word vector of the corresponding vocabulary.

For different users' legal consultation scenarios, issues that intention understanding algorithm of users' legal consulting statements focus on are different. For different types of users' legal consulting statements, we set the weight matrix which is used to complete the normalization of the original tensor representing user's legal consulting statement. The mathematical description of the normalization of the core tensor is as follows:

Given an original tensor $\widetilde{\chi}$ that representing user's legal consulting statement, $\chi \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, and a weight matrix W_m, $W_m \in \mathbb{R}^{I_2 \times I_1}$. The normalization of the original tensor $\widetilde{\chi}$ is expressed as

$$\widetilde{\chi} = \chi \times_S W_m \tag{4}$$

where $\times_S$ represents the operation of elongation or compression. $\widetilde{\chi} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, and the value of each element in υ is

$$\widetilde{\chi}(i_1, i_2, i_3) = \chi(i_1, i_2, i_3) W_m(i_2 . i_1) \tag{5}$$

For the problem of learning users' legal consulting intention, the weight matrix W_m contains the weight information of different modules in the user's legal consulting statement and the weight information of the words that need to be focused or ignored in each module. W_m is pre-set by developers according to scenarios of different users' legal consultation, and provides powerful interpretability and universality for the subsequent classification model based on Bi-LSTM upon users' legal consulting statements.

In this article, the Tucker tensor decomposition method [Cichocki, Zdunek and Amari (2008)] is used to decompose the normalization tensor, and the core tensor that can represent the original tensor element and structure information is obtained. The core tensor is then used in the subsequent Bi-LSTM algorithm to complete the classification of intention of users' legal consulting statements. The Tucker tensor decomposition method decomposes the original tensor into a core tensor and its corresponding set of factor matrices. The Tucker tensor decomposition method approximates the original tensor by calculating the n-mode product of the core tensor and the corresponding factor matrix on each mode. Its formal description is as follows:

Given an N-mode tensor β, $\beta \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. The Tucker tensor decomposition form of β is expressed as

$$\beta \approx \kappa \times_1 U^{(1)} \times_2 U^{(2)} \times \cdots \times_N U^{(N)} \tag{6}$$

That is

$$\beta \approx \kappa \prod_{n=1}^{N} \times_n U^{(n)} = \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} \kappa(i_1, \cdots, i_N) u_{i_1}^{(1)} \circ \cdots \circ u_{i_N}^{(N)} \tag{7}$$

where κ is the obtained core tensor, $\kappa \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$. $U^{(n)}$ is the corresponding factor matrices, $U^{(n)} \in \mathbb{R}^{J_n \times I_n}$. $u_{i_n}^{(n)}$ represents the i_n th column vector of $U^{(n)}$.

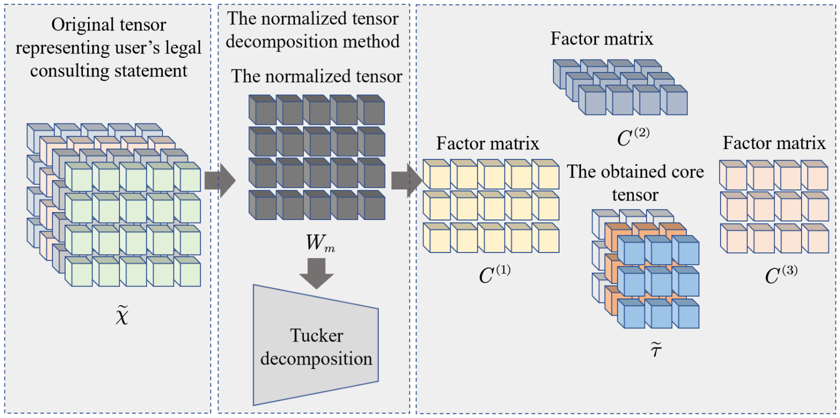

Figure 2: The framework of the normalized tensor decomposition method

As shown in Fig. 2, for the purpose of the classification of users' legal consulting intention, the normalized tensor decomposition method can be described as

$$\tilde{\chi} \approx \tilde{\tau} \times_1 C^{(1)} \times_2 C^{(2)} \times_3 C^{(3)} \tag{8}$$

That is

$$\tilde{\chi} \approx \tilde{\tau} \prod_{n=1}^{3} \times_n C^{(n)} = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \sum_{i_3=1}^{I_3} \tilde{\tau}(i_1, i_2, i_3) c_{i_1}^{(1)} \circ c_{i_2}^{(2)} \circ c_{i_3}^{(3)} \tag{9}$$

where $\tilde{\chi}$ is the normalized tensor of user's legal consulting statement χ, $\tilde{\chi} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$. $\tilde{\tau}$ is the core tensor obtained by the Tucker decomposition method with $\tilde{\chi}$ as input, $\tilde{\tau} \in \mathbb{R}^{J_1 \times J_2 \times J_3}$. $C^{(n)}$ is the corresponding factor matrices, $C^{(n)} \in \mathbb{R}^{J_n \times I_n}$. $c_{i_n}^{(n)}$ represents the i_n th column vector of $C^{(n)}$.

Compared to traditional Tucker tensor decomposition method, the normalized tensor decomposition sets a new parameter, namely the weight matrix W_m. The Tucker tensor decomposition method directly decomposes original tensors representing users' legal consulting statements into uninterpretable core tensors. It is an unsupervised behavior. However, for different types of users' legal consulting statements, the weight of each module in original tensors is different.

Algorithm 1 shows the pseudo-code of the normalized tensor decomposition method. In line 1, function $Tucker()$ corresponds to Eqs. (8) and (9). Line 3 corresponds to Eqs. (4) and (5).

Algorithm 1: The normalized tensor decomposition method for tensor represent the users' legal consulting statements

Input: Tensor $\widetilde{\chi}$ that represents the user's legal consulting statement, $\widetilde{\chi} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, and the weight matrix W_m, $W_m \in \mathbb{R}^{J_2 \times J_1}$.

Output: The normalized core tensor υ which carries the main tensor elements and tensor structure information of the original tensor $\widetilde{\chi}$ under the influence of the weight matrix W_m, $\upsilon \in \mathbb{R}^{J_1 \times J_2 \times J_3}$.

1 $\tilde{\tau} = Tucker(\widetilde{\chi})$;

2 **for** (j_1, j_2, j_3) in $Range(J_1, J_2, J_3)$ **do**

3 $\quad \upsilon(j_1, j_2, j_3) = \tilde{\tau}(j_1, j_2, j_3) W_m(j_2 . j_1)$;

4 **end**

5 **return** υ;

4.2 The TdBrnn neural network

This section details the intention understanding method of users' legal consulting statements, namely TdBrnn. As shown in Fig. 3, TdBrnn is a classification method based on normalized tensor decomposition and Bi-LSTM [Graves, Jaitly and Mohamed (2013)]. We represent the user's legal consulting statement as tensor χ. For different users' legal consultation scenarios, the focus of the problem is different. Based on the specific problem, we set the weight matrix W_m to normalize χ. Then we get the normalized tensor $\widetilde{\chi}$. We use the Tucker tensor decomposition method to decompose $\widetilde{\chi}$ into core tensor $\tilde{\tau}$. $\tilde{\tau}$ represents the main tensor structure and element information in $\widetilde{\chi}$.

The normalized tensor decomposition method is actually a principal component extraction operation. This operation not only reduces the tensor dimension, but also reduces the influence of the meaningless information in \widetilde{\chi} on the accuracy of the subsequent classification algorithm, and enhances the relative weight of the meaningful information. In turn, we use Bi-LSTM to classify obtained core tensors. The neural network is trained according to the principle of forward propagation and error back propagation. Finally, the intention of users' legal consultation is understood.

This next part describes the forward propagation and error back propagation principles of bidirectional LSTM used in this article.

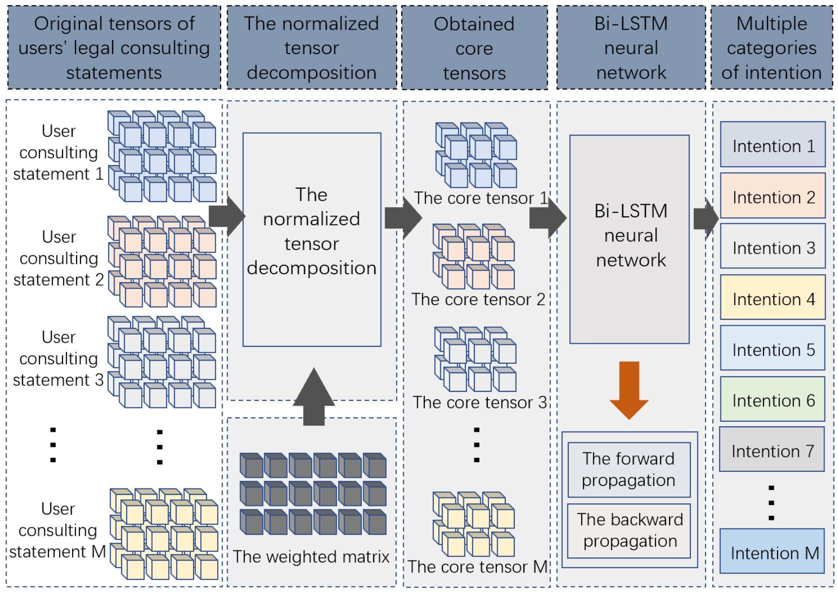

Figure 3: The framework of TdBrnn neural network

4.2.1 The forward propagation process of the Bi-LSTM

In this article, we use a bidirectional LSTM to classify the normalized core tensors which are obtained from the previous section and represent the users' legal consulting statements.

The bidirectional LSTM consists of a forward LSTM and a backward LSTM. The bidirectional LSTM neurons simultaneously store two states, the forward LSTM unit output state and the backward LSTM unit output state, and the final output of the bidirectional LSTM neuron is formed by connecting the two states in series.

The output calculation equations of the Bi-LSTM are as follows:

$$h_{f_t} = LSTM_{forward}(h_{f_{t-1}}, x_t) \tag{10}$$

$$h_{b_t} = LSTM_{backward}(h_{b_{t+1}}, x_t) \tag{11}$$

$$h_t = [h_{f_t}, h_{b_t}] \tag{12}$$

where $h_{f_{t-1}}$ is the output of the forward LSTM neuron at time $t-1$, h_{f_t} is the output of the forward LSTM neuron at time t, $h_{b_{t+1}}$ is the output of the backward LSTM neuron at time $t+1$ and h_{b_t} is the output of the backward LSTM neuron at time t. h_t is the output of Bi-LSTM neuron at time t. $LSTM_{forward}$ and $LSTM_{backward}$ are forward propagation algorithms for the forward LSTM and backward LSTM, respectively.

Eqs. (10), (11) and (12) embody the forward propagation process of a Bi-LSTM.

4.2.2 The error back propagation process of the Bi-LSTM along time and hidden layers

Since the bidirectional LSTM is composed of forward LSTM and backward LSTM, the forward LSTM and the backward LSTM have the same forward propagation formulas and error back propagation principle, so in this article, we use the forward LSTM as an example to introduce the error back propagation of the bidirectional LSTM along time and the hidden layers.

From Eq. (1), we can derive the weighted inputs of the hidden layer at time t in the LSTM. That is

$$\begin{bmatrix} net_{f,t} \\ net_{i,t} \\ net_{o,t} \\ net_{\tilde{c},t} \end{bmatrix} = \begin{bmatrix} w_{fh}, w_{fx} \\ w_{ih}, w_{ix} \\ w_{oh}, w_{ox} \\ w_{ch}, w_{cx} \end{bmatrix} \cdot \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} + \begin{bmatrix} b_f \\ b_i \\ b_o \\ b_c \end{bmatrix} \tag{13}$$

where $net_{f,t}$, $net_{i,t}$, $net_{o,t}$ and $net_{\tilde{c},t}$ are the weighted inputs of the corresponding gates, respectively.

We represent the loss function that is used in the Bi-LSTM with $lossF$. Under the premise of knowing ρ_t, $\rho_t = \frac{\partial lossF}{\partial h_t}$, what we need to do is to find the value of ρ_{t-1}, $\rho_{t-1} = \frac{\partial lossF}{\partial h_{t-1}}$.

From the full derivative formula of functions, we can draw that

$$\rho_{t-1} = \frac{\partial lossF}{\partial h_t}\frac{\partial h_t}{\partial h_{t-1}} \tag{14}$$

From Eqs. (1), (2), (3), (13) and (14), according to the full derivative formula of functions, we can deduce the following formula:

$$\begin{aligned} \rho_{t-1} &= \rho_t \frac{\partial h_t}{\partial o_t}\frac{\partial o_t}{\partial net_{o,t}}\frac{\partial net_{o,t}}{\partial h_{t-1}} + \rho_t \frac{\partial h_t}{\partial c_t}\frac{\partial c_t}{\partial f_t}\frac{\partial f_t}{\partial net_{f,t}}\frac{\partial net_{f,t}}{\partial h_{t-1}} \\ &+ \rho_t \frac{\partial h_t}{\partial c_t}\frac{\partial c_t}{\partial i_t}\frac{\partial i_t}{\partial net_{i,t}}\frac{\partial net_{i,t}}{\partial h_{t-1}} + \rho_t \frac{\partial h_t}{\partial c_t}\frac{\partial c_t}{\partial \widetilde{c_t}}\frac{\partial \widetilde{c_t}}{\partial net_{\tilde{c},t}}\frac{\partial net_{\tilde{c},t}}{\partial h_{t-1}} \end{aligned} \tag{15}$$

Further, we can derive the following simplified formula of Eq. (15) for calculating ρ_{t-1}.

$$\rho_{t-1} = \rho_{o,t} w_{oh} + \rho_{f,t} w_{fh} + \rho_{i,t} w_{ih} + \rho_{\tilde{c},t} w_{ch} \tag{16}$$

where $\rho_{o,t} = \rho_t * \tanh(c_t) * o_t * (1 - o_t)$, $\rho_{f,t} = \rho_t * o_t * (1 - \tanh(c_t)^2) * c_{t-1} * f_t * (1 - f_t)$, $\rho_{i,t} = \rho_t * o_t * (1 - \tanh(c_t)^2) * \widetilde{c_t} * i_t * (1 - i_t)$, $\rho_{\tilde{c},t} = \rho_t * o_t * (1 - \tanh(c_t)^2) * i_t * (1 - \widetilde{c_t}^2)$.

By the same token, the propagation formula of the error between different hidden layers is derived as follows:

$$\rho_t^{l-1} = (\rho_{o,t}^l w_{ox} + \rho_{f,t}^l w_{fx} + \rho_{i,t}^l w_{ix} + \rho_{\tilde{c},t}^l w_{cx}) * \frac{\partial activef}{\partial net_t^{l-1}} \tag{17}$$

where $activef$ is the active function of the $i-1$th hidden layer. ρ_t^{l-1} is the error of the $i-1$th hidden layer. $\rho_{f,t}^l$, $\rho_{i,t}^l$, $\rho_{o,t}^l$ and $\rho_{\tilde{c},t}^l$ are the error terms of the lth hidden layer to the corresponding gates at time t.

5 Empirical results

This chapter gives the experimental results and analysis in this article. Section 5.1 provides the statistical information of dataset contains users' legal consulting statements and related parameter settings. Section 5.2 gives the parameter configuration of neural networks used in the experiment part of this article. Section 5.3 shows the experimental comparison and result analysis of TdBrnn with other recurrent neural networks.

5.1 Data description and preprocessing operations

The experimental data used in this article is real data crawled from the legal advisory service platform in various parts of China. This includes nearly 300,000 pieces of data that have been tagged by legal staff with backgrounds in professional legal knowledge, involving 40 types of legal advice. Fig. 4 provides the distribution of partial types of users' legal consulting statements.

It should be clarified that the original tensors representing the users' legal consulting statements must have the same dimensions, so that the corresponding core tensors generated by the tensor decomposition layer have the same dimensions, and thus serve as inputs to the bidirectional LSTM. Fig. 5 shows the length distribution of users' legal consulting statements in the data set used here. In this article we use 40 as the uniform length of the users' legal consulting statement. We use the vocabulary word vectors generated in the previous step and the users' legal consulting statement modules to generate the original tensors representing the users' legal consulting statements.

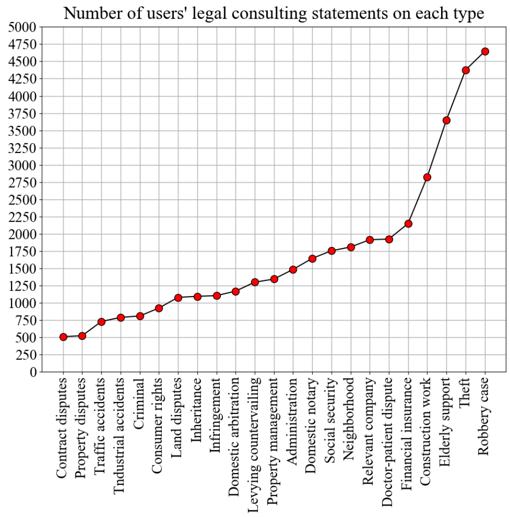

Figure 4: The distribution of users' legal consulting statements on each type

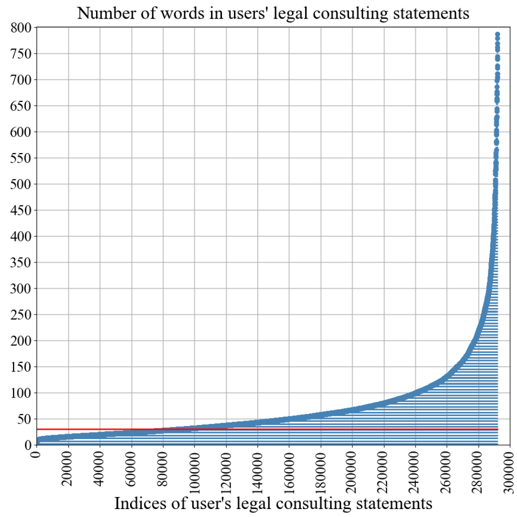

Figure 5: Length of users' legal consulting statements

To use the users' legal consulting statements as the inputs of the Bi-LSTM and perform hyperparameter training, multiple data preprocessing operations are needed, such as Chinese vocabulary participles, removal of stop words and Chinese punctuation marks, removal of meaningless and redundant information, etc. Ultimately we generate word vectors for all the remaining words. The word vector generation operation is a crucial step in the data preprocessing process. This article uses Google's word2vec tool [Zhang and Lecun (2015)] to generate Chinese word vectors model for intention understanding of users' legal consultation. The corpus used is Sogou Lab data, Tencent News, Sina Weibo, Chinese Wikipedia and legal cases in China.

5.2 Experimental configuration and parameter adjustment

In this article, we present a method for the classification of intention in users' legal consulting statements, namely TdBrnn. TdBrnn combines Bi-LSTM with the normalized tensor decomposition method, which extracts the main structure and elements information from original tensors. We use the tensorflow development kit [Abadi, Barham, Chen et al. (2016)] to implement neural networks involved in our experiment. Simultaneously, in order to save the running time and improve the running speed of programs, our experiment is performed on the Graphics Processing Unit (GPU) device.

Different from the parameter configuration of traditional neural networks, in addition to the size of batches, the number of hidden layers, the size of hidden layers, the value of learning rate and the number of iterations, parameters involved in TdBrnn contain weight matrix W_m. W_m is an important parameter in the normalized tensor decomposition method. For different legal issue backgrounds, each module in users' legal consulting statements has different weights. The weight matrix in this article is pre-set according to subjects of users' legal consulting statements.

For all neural networks used in our experiment, including RNN, LSTM, GRU, Bi-LSTM, Bi-GRU and TdBrnn in Section 5.3, we set the size of batches to 30, the number of hidden layers to 3, the size of hidden layers to 512, the value of learning rate to 0.001 and the number of iterations to 10.

5.3 Experimental results and analysis

This section gives the performance of TdBrnn in the classification of intention in users' legal consulting statements. Section 5.3.1 provides a comparison experiment between TdBrnn and other commonly used neural networks, such as RNN [Zaremba, Sutskever and Vinyals (2014)], LSTM [Wang and Jiang (2016)], GRU [Dey and Salemt (2017)], Bi-LSTM [Zhao, Yan, Wang et al. (2017)] and Bi-GRU [Liu, Wang, Liu et al. (2017)]. In addition, TdBrnn proposed in this paper is a deep learning algorithm based on Bi-LSTM and normalized tensor decomposition. In order to demonstrate the superiority of Bi-LSTM over other neural networks, this paper conducted a more in-depth comparative experiment from the perspective of normalized tensor decomposition methods. Section 5.3.2 gives the performance of neural networks with and without normalized tensor decomposition layer in the classification of intention in users' legal consulting statements.

5.3.1 Experimental results and analysis of TdBrnn and other neural networks

In this article, Bi-LSTM with tensor decomposition layer for users' legal consulting intent understanding is compared with various recurrent neural networks, such as bidirectional LSTM without tensor decomposition layer, RNN, unidirectional LSTM, GRU and Bi-GRU. The experimental results are shown in Fig. 6.

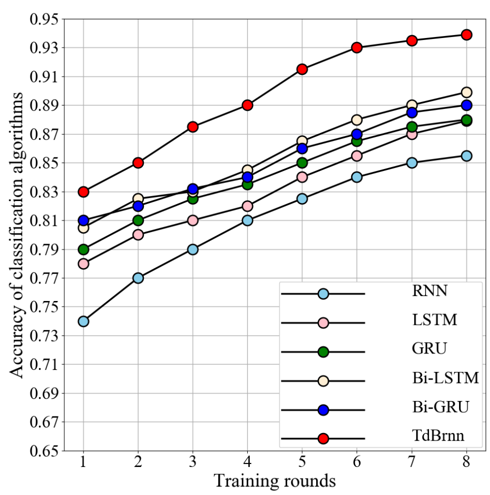

Figure 6: Accuracy of neural networks on the classification of intention in users' legal consulting statements

As can be seen from Fig. 6, our proposed method TdBrnn with normalized tensor decomposition layer is superior to other recurrent neural networks without tensor decomposition layer in our experiments.

The main issue is the use of the normalized tensor decomposition layer. The normalized tensor decomposition layer extracts the tensor structure and elements information that is most beneficial to the classification of intention in the users' legal consulting statements. It reduces the influence of useless information on the training process of hyper parametric of deep neural network classification models. And it relatively increases the weight of important information related to legal elements.

Traditional RNNs are prone to gradient disappearance or explosion when dealing with long-distance dependence problems. This phenomenon seriously affects the accuracy of classification models. LSTMs solve the above problem by setting gates and long-term unit status. Therefore, compared with RNNs, LSTMs have higher accuracy in classifying users' legal consulting statements.

Bi-LSTM utilizes forward and backward LSTMs to comprehensively utilize contextual information in text sequences. Thus, Bi-LSTM provides more complete analysis of consulting data and higher classification accuracy than LSTM. Bi-GRU and GRU are the same principle. Compared with LSTMs, the number of parameters in GRUs is greatly reduced. The convergence speed of GRUs is faster. However, the accuracy of LSTMs and LSTMs is comparable.

Compared with other RNNs, the novelty of TdBrnn is the application of normalized tensor decomposition method, which extracts the main tensor structure and elements information from the original tensor. This information is useful for improving the accuracy of the subsequent classification model. The normalized tensor decomposition method reduces the dimensions of original tensors and greatly reduces the computational complexity of the subsequent classification algorithm.

5.3.2 Experimental results and analysis of neural networks with and without normalized tensor decomposition method

The normalized tensor decomposition method is one of the important innovations in this article. In order to better highlight the advantages of the combination of the normalized tensor decomposition method and Bi-LSTM, we conducted a more in-depth comparative experiment. In this section we combine the normalized tensor decomposition method with a variety of neural networks.

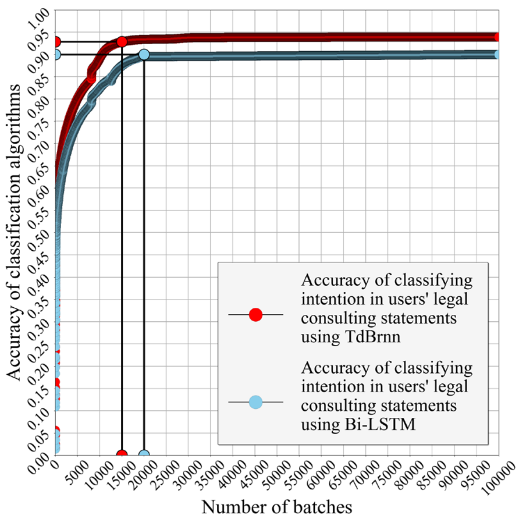

Figure 7: Accuracy of TdBrnn and Bi-LSTM on the classification of intention in users' legal consultation statements

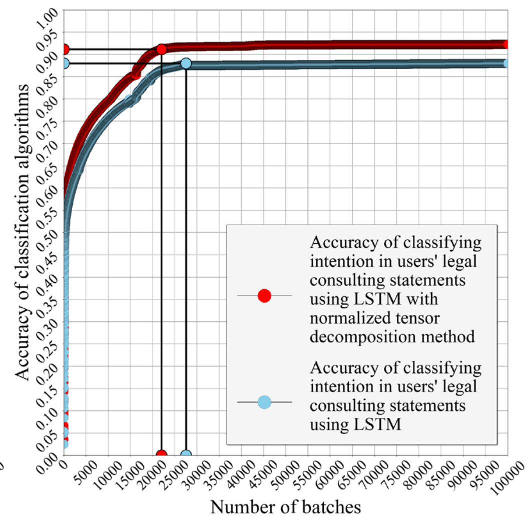

Figure 8: Accuracy of LSTM with and without normalized tensor decomposition method

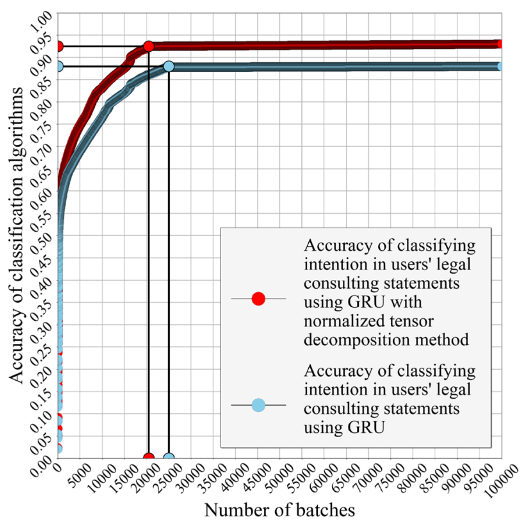

Figure 9: Accuracy of GRU with and without normalized tensor decomposition method

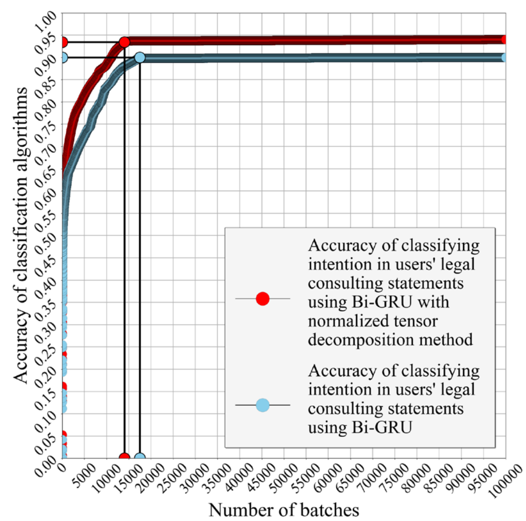

Figure 10: Accuracy of Bi-GRU with and without normalized tensor decomposition method

Fig. 7 shows the classification accuracy of TdBrnn and Bi-LSTM in the understanding of users' legal consulting intention. In essence, TdBrnn adds a normalized tensor decomposition layer to Bi-LSTM. As can be seen from Fig. 7, TdBrnn has a higher accuracy than Bi-LSTM. In addition, as the number of batches increases, TdBrnn reaches convergence faster. The same phenomenon can be seen in Figs. 8-11.

The normalized tensor decomposition layer extracts the main tensor structure and element information from original tensors representing users' legal consulting statements. This operation is equivalent to a primary component extraction, which not only reduces the tensor dimension, but also removes redundant, meaningless information from the original tensor. Core tensors obtained by the normalized tensor decomposition method contain information that is most conducive to improving the accuracy of the subsequent classification method. Therefore, neural networks with the normalized tensor decomposition layer have higher accuracy and faster convergence in the classification of users' legal consulting intentions than that without.

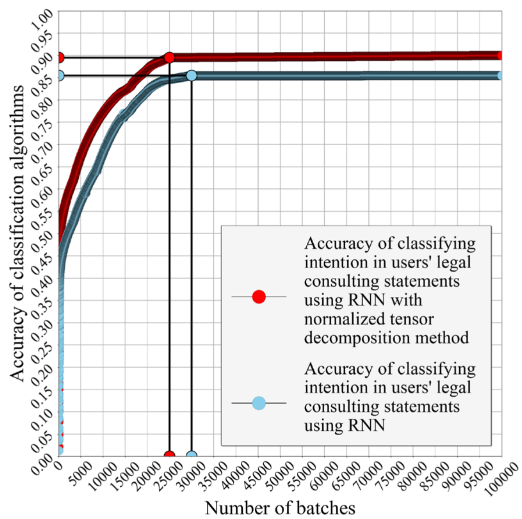

Figure 11: Accuracy of RNN with and without normalized tensor decomposition method

The normalized tensor decomposition method greatly reduces the computational complexity of original algorithms. It indirectly reduces the impact of noise information on the classification algorithm and enhances the weight of meaningful information. Figs. 7-11 fully demonstrate the role of the normalized tensor decomposition method in the classification of intention in users' legal consulting statements.

As can be seen from Figs. 8 and 9, LSTM and GRU are comparable in classification accuracy, as well as with a standardized tensor decomposition layer. But in terms of convergence speed, GRU is obviously better. The main factor is that GRU has been further streamlined on the basis of LSTM. GRU combines the input gate, the forget gate, and the output gate into two gates, namely the update gate and the reset gate. Moreover, GRU combines the unit state and the output state into one state. The number of hyperparameters in GRU is much lower than LSTM. Therefore, GRU has lower computational complexity and faster convergence, but maintains similar accuracy with LSTM.

From Fig. 7 and 8, we can see that algorithms based on Bi-LSTM have higher accuracy than algorithms based on LSTM for the classification of intention in users' legal consulting statements. Bi-LSTM consists of the forward LSTM and the backward LSTM. The forward LSTM carries the past context information of the sequence. The backward LSTM carries future context information. Combined with the two, Bi-LSTM can more accurately understand users' legal consulting intention. The same applies to GRU and Bi-GRU. As can be seen from Fig. 9 and Fig. 10, Bi-GRU has a higher accuracy with respect to GRU. As the number of batches grows, Bi-GRU reaches a convergence state earlier.

Fig. 11 shows the performance of RNN and RNN with normalized tensor decomposition layer in the classification of intention in users' legal consulting statements. It can be seen that the accuracy of RNNs is significantly reduced and the convergence speed is slower than that of LSTMs and GRUs. The main reason is that traditional RNN is prone to gradient explosion or gradient disappearance when dealing with long-distance dependence problems. LSTMs and GRUs solve the above problems by setting gates and long-term units. Therefore, LSTMs and GRUs perform better than RNNs in classifying users' legal consulting statements.

6 Conclusion

This article proposes a new method for intention understanding of users' legal consultation TdBrnn, which is based on normalized tensor decomposition method and Bi-LSTM. TdBrnn represents users' legal consulting statements as tensors. And then extracts core tensors from original tensors using the normalized tensor decomposition method. TdBrnn trains Bi-LSTM using obtained core tensors, and finally completes the classification of intention in users' legal consulting statements.

The normalized tensor decomposition method extracts the main structure and element information in original tensors which represent users' legal consulting statements. It is actually equivalent to a principal component extraction. The normalized tensor decomposition method reduces the tensor dimension of original tensors and reduces the computational complexity of the subsequent Bi-LSTM. It reduces the impact of meaningless information on the classification model while relatively enhanced weighting of meaningful information.

However, our algorithm still needs improvement. In the future work, we focus on improving the speed of calculation and the accuracy of our algorithm in order to understand the intention of users' legal consultation in a more accurate and faster way.

Acknowledgement: Thanks to Network and Information Security Research Center in Harbin Institute of Technology for their support of this article. This work is supported by the National Key Research and Development Program of China (2018YFC0830602, 2016QY03D0501), National Natural Science Foundation of China (61872111).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A. et al. (2016): Tensorflow: a system for large-scale machine learning. https://arxiv.org/pdf/1605.08695.pdf.

Capuano, N.; Maio, C. D.; Salerno, S.; Toti, D. (2014): A methodology based on commonsense knowledge and ontologies for the automatic classification of legal cases. *International Conference on Web Intelligence.*

Chen, J.; Huang, H.; Tian, S.; Qu, Y. (2009): Feature selection for text classification with Naïve Bayes. *Expert Systems with Applications*, vol. 36, no. 3, pp. 5432-5435.

Cichocki, A.; Zdunek, R.; Amari, S. I. (2008): Nonnegative matrix and tensor factorization [lecture notes]. *IEEE Signal Processing Magazine*, vol. 25, no. 1, pp. 142-145.

De Lathauwer, L. (2009): Tensor decompositions and applications: a survey.

https://lirias.kuleuven.be/handle/123456789/375086.

Dey, R.; Salemt, F. M. (2017): Gate-variants of gated recurrent unit (GRU) neural networks. *IEEE 60th International Midwest Symposium on Circuits and Systems*, pp. 1597-1600.

Galgani, F.; Compton, P.; Hoffmann, A. (2015): LEXA: building knowledge bases for automatic legal citation classification. *Expert Systems with Applications*, vol. 42, no. 17-18, pp. 6391-6407.

Greff, K.; Srivastava, R. K.; Koutník, J.; Steunebrink, B. R.; Schmidhuber, J. (2015): LSTM: a search space odyssey. *IEEE Transactions on Neural Networks & Learning Systems*, vol. 28, no. 10, pp. 2222-2232.

Gers, F. A.; Schmidhuber, J.; Cummins, F. (2000): Learning to forget: continual prediction with LSTM. *Neural Computation*, vol. 12, no. 10, pp. 2451-2471.

Graves, A.; Jaitly, N.; Mohamed, A. R. (2013): Hybrid speech recognition with deep bidirectional LSTM. *IEEE Workshop on Automatic Speech Recognition and Understanding*.

Kanapala, A.; Pal, S.; Pamula, R. (2017): Text summarization from legal documents: a survey. *Artificial Intelligence Review.* https://doi.org/10.1007/s10462-017-9566-2.

Liu, W.; Wang, Z.; Liu, X.; Zeng, N.; Liu, Y. et al. (2017): A survey of deep neural network architectures and their applications. *Neurocomputing*, vol. 234, pp. 11-26.

Mcneice, G. W.; Jones, A. G. (2001): Multisite, multifrequency tensor decomposition of magnetotelluric data. *Geophysics*, vol. 66, no. 1, pp. 158-173.

Sulea, O. M.; Zampieri, M.; Malmasi, S.; Vela, M.; Dinu, L. P. et al. (2017): Exploring the use of text classification in the legal domain.

https://www.researchgate.net/publication/320619861_Exploring_the_Use_of_Text_Classification_in_the_Legal_Domain.

Sidiropoulos, N.; De Lathauwer, L.; Fu, X.; Huang, K.; Papalexakis, E. et al. (2017): Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551-3582.

Wang, S.; Jiang, J. (2016): Learning natural language inference with LSTM. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.*

https://ieeexplore.ieee.org/document/8053243.

Zhang, T.; Oles, F. J. (2001): Text categorization based on regularized linear classification methods. *Information Retrieval*, vol. 4, no. 1, pp. 5-31.

Zhang, W.; Yoshida, T.; Tang, X. (2008): Text classification based on multi-word with support vector machine. *Knowledge-Based Systems*, vol. 21, no. 8, pp. 879-886.

Zhang, X.; Lecun, Y. (2015): Text understanding from scratch. *Computer Science.*

Zaremba, W.; Sutskever, I.; Vinyals, O. (2014): Recurrent neural network regularization. arXiv: 1409.2329v5.

Zhao, R.; Yan, R.; Wang, J.; Mao, K. (2017): Learning to monitor machine health with convolutional bi-directional LSTM networks. *Sensors*, vol. 17, no. 2, pp. 273.