

# 基于模板张量分解和双向 LSTM 的 司法案件罪名认定

李大鹏<sup>1,2,3</sup>, 陈剑<sup>2,3</sup>, 王晨<sup>2,3</sup>, 闻英友<sup>2,3</sup>, 赵大哲<sup>1</sup>

(1. 东北大学计算机科学与工程学院, 辽宁沈阳 110819; 2. 东北大学东软研究院, 辽宁沈阳 110819;  
3. 辽宁省工业控制安全技术工程中心, 辽宁沈阳 110819)

**摘要:** 案件罪名认定是司法业务的重要环节, 尚缺乏有效的智能辅助工具和手段. 针对案件定罪的难点问题, 提出一种结合张量分解和双向 LSTM (Long Short-Term Memory) 神经网络的案件定罪方法. 该方法将案件数据表示为张量, 并在张量分解过程中引入模板张量. 模板张量可以在双向 LSTM 神经网络分类模型的训练过程不断的被优化, 使得分解后的核心张量包含更加有效的张量结构和特征信息, 有助于提高后续分类模型的准确性, 实现案件罪名的精准认定. 实验结果表明: 所提出的基于张量分解和双向 LSTM 的司法案件定罪方法比现有方法具有更好的准确性.

**关键词:** 张量分解; 双向长短期记忆; 模板张量; 案件定罪; 文本分类

**中图分类号:** TP183 **文献标识码:** A **文章编号:** 0372-2112 (2021)04-0760-08

**电子学报 URL:** <http://www.ejournal.org.cn> **DOI:** 10.12263/DZXB.20200098

## Conviction in Judicial Cases Based on Template Tensor Decomposition and Bidirectional LSTM

LI Da-peng<sup>1,2,3</sup>, CHEN Jian<sup>2,3</sup>, WANG Chen<sup>2,3</sup>, WEN Ying-you<sup>2,3</sup>, ZHAO Da-zhe<sup>1</sup>

(1. School of Computer Science and Engineering, Northeastern University, Shenyang, Liaoning 110819, China;

2. Neusoft Group Research, Northeastern University, Shenyang, Liaoning 110819, China;

3. Research Center of Safety Engineering Technology in Industrial Control of Liaoning Province, Shenyang, Liaoning 110819, China)

**Abstract:** Conviction in judicial cases is an important part of judicial business, but there is still a lack of effective intelligent auxiliary tools and methods. Aiming at the difficult problem of conviction in judicial cases, a method combining tensor decomposition and Bi-LSTM neural network is proposed. This method represents the case data as a tensor and introduces a template tensor in the tensor decomposition process. The template tensor can be continuously optimized during the training process of Bi-LSTM neural network classification model, so that the decomposed core tensor contains more effective tensor structure and feature information, which is helpful to improve the accuracy of the subsequent classification model and realize the accurate conviction in judicial cases. The experimental results show that the proposed method for conviction in judicial cases based on tensor decomposition and Bi-LSTM has better accuracy than the existing methods.

**Key words:** tensor decomposition; Bi-LSTM; template tensor; conviction in judicial cases; text classification

## 1 引言

近年来, 随着社会经济快速发展以及依法治国战略不断推进, 新一代信息技术在司法领域的应用受到了广泛关注<sup>[1]</sup>. 案件定罪是指司法机构依据刑法规定, 对某一危害行为是否构成犯罪, 是何种犯罪的认定过程. 在当前的司法实践中, 案件的定罪工作普遍依靠人工分析案件卷宗后通过经验进行认定, 尚缺乏有效的

智能辅助工具和手段. 研究司法案件定罪智能方法, 有效解决定罪环节中的难点问题, 对我国公正司法、司法为民的司法体系建设具有重要意义.

司法办案人员需要通过犯罪事实和讯问笔录等案件卷宗内容来进行罪名认定工作. 在实际的司法实践中, 案情较为复杂、存在数罪并罚和罪名争议的案件是定罪工作的难点. 尤其是具有罪名争议的案件, 往往量刑上区别很大, 例如非法吸收公众存款罪与集资诈骗

收稿日期: 2020-01-15; 修回日期: 2020-09-22; 责任编辑: 马兰英

基金项目: 国家自然科学基金 (No. 61972079, No. 61772126); 国家重点研发计划 (No. 2018YFC0830601); 教育部基本科研业务费 (No. 171802001, No. 2016002, No. 2016004); 辽宁省重点研发计划 (No. 2019JH2/10100027); 辽宁省“兴辽英才”计划项目 (No. XLYC1802100)

罪具有相似的行为表现方式,犯罪行为描述也相似,办案人员需要围绕争议焦点进行仔细的案情分析和判断.本文将聚焦相似罪名定罪的难点问题开展研究工作.

在司法领域,机器学习尤其是深度学习的应用成为了智慧司法技术发展的新趋势.司法案件的定罪可以归结为一个文本分类问题,即通过分析案件文本信息,为该案件确定一个或多个类别. Octavia-Maria 等人将支持向量机用于法律案件的分类<sup>[2]</sup>,并提出了一种结合统计和机器学习的方法用于预测法国最高法院的判决结果<sup>[3]</sup>.林志宏等人利用卷积神经网络提取案件文本中的语义特征<sup>[4]</sup>,从而支持对案件数据的深度挖掘. Xiao 等人使用多任务 CNN 网络模型完成中文法律咨询问题的分类<sup>[5]</sup>.上述研究的文本建模方法是基于特征标签的,标记特征需要大量的手工工作和专家知识.因此,在此基础上形成的案件分类算法不具有可扩展性. Guo 等人首次使用张量分解和深度神经网络分析法律咨询问题从而预测法律咨询者的意图<sup>[6]</sup>.使用张量对法律文本建模,可以将存在于案件文本中的关键信息具有的长程依赖转变为某一维度下的短程依赖,通过张量模型可以更加容易分析出文本前后内容的内在联系,但传统的张量分解方法<sup>[7]</sup>在得到核心张量的过程中会造成一定的信息缺失,影响后续的文本分类结果.

为了解决上述案件定罪研究的缺点,本文从两方面开展工作:(1)对于案件数据的建模,本文提出一种基于模板的张量分解(Template-based Tensor Decomposition, TTD)算法,相对于传统的张量分解算法,该算法能够最大限度的保留初始张量内的关键信息和张量结构,生成准确表征案件数据的低阶张量;(2)对于案件的分类,我们选择使用长短期记忆网络(Bidirectional Long Short-Term Memory, Bi-LSTM)分类模型<sup>[8]</sup>结合 TTD 算法,实现案件罪名的精准分类.

## 2 背景知识

### 2.1 张量基本知识

张量是高维数组的泛概念,举例来说,一阶张量我们通常称之为向量,二阶张量是矩阵,三阶张量或者更高阶的张量称为高阶张量.

**定义 1 模式(mode)** 张量的模式定义为张量的维度大小.例如,张量  $\chi \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  的模式为  $N$ , 又称为阶.

**定义 2 Frobenius 范数(Frobenius norm)** 给定一个  $N$  阶张量  $\chi \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ , 张量的 Frobenius 范数记为  $\|\chi\|_F$ :

$$\|\chi\|_F = \sqrt{\sum_{i_1}^{I_1} \dots \sum_{i_N}^{I_N} \chi(i_1, \dots, i_N)^2} \quad (1)$$

**定义 3  $n$  模式乘积( $n$ -mode product)**  $n$  模式乘积为张量  $\chi \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  和矩阵  $U \in \mathbb{R}^{I_n \times J}$  在第  $n$  模式上的乘积,可以表示为  $\lambda = \chi \times_n U, \lambda \in \mathbb{R}^{I_1 \times \dots \times I_{n-1} \times J \times I_{n+1} \times \dots \times I_N}$ .

$$\begin{aligned} \lambda(i_1, \dots, i_{n-1}, j, i_{n+1}, \dots, i_N) \\ = \sum_{m=1}^{I_n} \chi(i_1, \dots, i_{n-1}, m, i_{n+1}, \dots, i_N) U(m, j) \end{aligned} \quad (2)$$

### 2.2 张量分解

张量分解从本质上来说是矩阵分解的高阶泛化.常见的包括 Tucker 模型<sup>[9]</sup>和 CP 模型<sup>[10]</sup>. Tucker 模型是一种高阶主成分分析方法,它将  $N$  阶张量  $\chi \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  分解为一个核心张量(core tensor)和  $N$  个因子矩阵乘积形式:

$$\chi \approx \eta \times_1 U_1 \times_2 U_2 \dots \times_N U_N \quad (3)$$

其中,  $U_n \in \mathbb{R}^{J \times I_n}$ , 我们用  $\{U_n\}$  表示因子矩阵的集合;核心张量  $\eta \in \mathbb{R}^{J \times J \times \dots \times J}$ , 我们可以将核心张量  $\eta$  看作是对原始张量  $\chi$  的压缩.张量分解方法就是要最小化目标函数  $\varphi$ :

$$\varphi = \left\| \chi - \eta \prod_n \times_n U_n \right\|_F \quad (4)$$

## 3 司法案件定罪方法

### 3.1 司法案件的张量表示

在已有的相关研究中,案件文本特征表示方法主要包括基于词语统计和基于向量空间模型两类方法.目前常用的方法是后者,即使用 word2vec 等工具将分词后的文本转为一组实数值向量的集合.但对于相似罪名定罪场景,办案人员往往需要仔细阅读该案件的起诉意见书(包含侦查机关认定的犯罪事实)、询问笔录(包含嫌疑人陈述的犯罪事实)、审查报告(包含证据完整性分析等内容)等多类卷宗并进行事实比对和证据分析.如果使用传统的文本特征表示方法,将上述文本整合在一起向量化后作为算法输入,相关的文本内容已经相距较远,即使上下文相关的分类模型也无法取得较好的效果.

如图 1 所示,本文使用一个三阶张量模型代表案件文本,第一个维度表示描述案件犯罪事实的基本组成部分,包括嫌疑人对犯罪事实的陈述笔录、侦查机关认定的犯罪事实等.第二个维度是每个组成部分中对案件定罪有意义的词汇.第三个维度是每个词的词向量.案件文本的张量表示方法能够将来自不同文档的案件犯罪事实文本内容关联起来,使其包含尽量多的关键信息.但引入张量也使得案件文本的表征形式更加的复杂,张量的行数、列数及深度往往很高,如词向量维数往往在 200 ~ 400 之间,需要使用张量分解算法降低维

数,降低计算复杂度,同时去掉冗余信息.

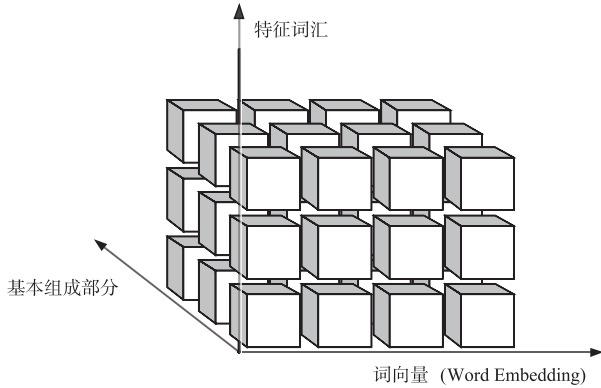


图1 案件的张量表示

### 3.2 方法基本框架

方法基本框架如图2所示,该方法主要分为两个步骤:(1)基于模板的张量分解(TTD),TTD算法将初始张量 $\chi$ 分解为核心张量 $\tilde{\chi}$ 和因子矩阵集合.该算法引入模板张量 $\nu$ ,其分解过程的目的是求得 $\tilde{\chi}$ 与 $\nu$ 最大化的近似;(2)Bi-LSTM神经网络分类模型,我们将分解后的核心张量输入分类模型,将案件数据映射到正确的罪名标签.同时,Bi-LSTM模型在网络训练过程中不断的优化模板张量 $\nu$ ,从而使张量分解后的核心张量 $\tilde{\chi}$ 包含更加有效的特征信息,提高后续Bi-LSTM分类模型的准确性.

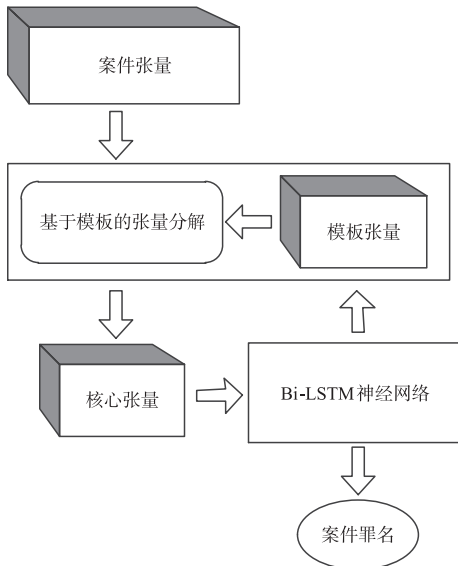


图2 方法框架

### 3.3 基于模板的张量分解算法

TTD算法的输入是案件张量 $\chi$ 和模板张量 $\nu$ ,分解过程将案件张量 $\chi$ 分解为核心张量 $\tilde{\chi}$ 和因子矩阵集 $\{C_k\}$ 和 $\{D_k\}$ ,并且使 $\tilde{\chi}$ 越来越近似模板张量 $\nu$ ,即 $\tilde{\chi}$ 与 $\nu$ 包含相似的张量结构.该算法可以理解为一个 $\chi$ 到 $\tilde{\chi}$

的映射函数,在有效降低张量维数的同时,从案件初始张量 $\chi$ 中提取用于案件定罪的重要信息.

**问题1** 给定一个张量 $\alpha \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ 和模板张量 $\nu \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_K}$ ,要求求得两个因子矩阵集 $\{C_k\}$ 和 $\{D_k\}$ , $C_k \in \mathbb{R}^{I_1 \times H_k}$ , $D_k \in \mathbb{R}^{J_1 \times H_k}$ ,同时 $\{C_k\}$ 和 $\{D_k\}$ 同时满足如下两个条件:

**条件1** 因子矩阵集 $\{C_k\}$ 和 $\{D_k\}$ 需要最小化目标函数:

$$\psi = \left\| \chi \prod_k C_k - \tilde{\chi} \prod_k D_k \right\|_F \quad (5)$$

**条件2** 因子矩阵集 $\{C_k\}$ 和 $\{D_k\}$ 中的矩阵都是正交矩阵.

为了计算 $\{C_k\}$ 和 $\{D_k\}$ ,引入交替最小二乘法(Alternating Least Squares, ALS)求得目标函数 $\phi$ 的最优解.求解式(5)需要使用如下引理:

**引理1** 张量 $\alpha \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ ,定义函数 $tr(\alpha^T \alpha) = \sum_{i_1} \sum_{i_2} \dots \sum_{i_N} \alpha(i_1, i_2, \dots, i_N)^2$ ,给定目标函数 $\delta = tr\left(\left(\chi \prod_k C_k\right)^T \left(\chi \prod_k C_k\right)\right)$ ,其中 $\chi \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ , $C_k \in \mathbb{R}^{I_1 \times H_k}$ ,且 $C_k$ 满足条件2,则针对矩阵 $C_{k_0}$ , $k_0 \in [1, K]$ ,目标函数 $\delta$ 对 $C_{k_0}$ 的偏导数 $\partial \delta / \partial C_{k_0} = \varepsilon \left(\chi \prod_{k \neq k_0} C_k\right)^T \left(\chi \prod_k C_k\right)$ ,其中 $\varepsilon$ 是一个常数.

**证明** 使用 $\mu$ 代表 $\chi \prod_{k \neq k_0} C_k$ ,则目标函数 $\delta$ 可表示成 $\delta = tr\left(\left(\mu \times_{k_0} C_{k_0}\right)^T \left(\mu \times_{k_0} C_{k_0}\right)\right)$ . $\delta$ 可缩写成 $\delta = tr\left(\left(\mu C_{k_0}\right)^T \left(\mu C_{k_0}\right)\right) = tr\left(C_{k_0}^T \mu^T \mu C_{k_0}\right)$ .使用 $\eta$ 表示 $\mu^T \mu$ ,则 $\delta = tr\left(C_{k_0}^T \eta C_{k_0}\right)$ ,根据函数求导规则, $\partial \delta / \partial C_{k_0} = \partial tr\left(C_{k_0}^T \eta C_{k_0}\right) / \partial C_{k_0} = \eta C_{k_0} + \eta^T C_{k_0}$ .而 $\eta = \eta^T$ ,所以目标函数 $\delta$ 对 $C_{k_0}$ 的偏导数 $\partial \delta / \partial C_{k_0} = 2\eta C_{k_0} = \varepsilon \left(\chi \prod_{k \neq k_0} C_k\right)^T \left(\chi \prod_k C_k\right)$ ,证毕.

我们使用 $\lambda$ 和 $\omega$ 表示 $\chi \prod_{k \neq k_0} C_k$ 和 $\nu \prod_k D_k$ ,则式(5)可写成 $\varphi = tr\left(\left(\lambda \times_{k_0} C_{k_0} - \omega\right)^T \left(\lambda \times_{k_0} C_{k_0} - \omega\right)\right)$ ,简称为 $\varphi = tr\left(\left(\lambda C_{k_0} - \omega\right)^T \left(\lambda C_{k_0} - \omega\right)\right)$ .对函数 $\varphi$ 求 $C_{k_0}$ 的偏导数 $\partial \varphi / \partial C_{k_0} = \left(\left(\partial tr\left(\omega^T \omega\right) - 2 \partial tr\left(C_{k_0}^T \lambda^T \omega\right) + \partial tr\left(C_{k_0}^T \lambda^T \lambda C_{k_0}\right)\right) / \partial C_{k_0}\right)$ ,根据函数求导规则, $\partial tr\left(\omega^T \omega\right) / \partial C_{k_0} = 0$ , $\partial tr\left(C_{k_0}^T \lambda^T \omega\right) / \partial C_{k_0} = \lambda^T \omega$ ,将其和引理1带入上式,得到:

$$C_{k_0}^T G_{k_0} = \left(\lambda C_{k_0}\right)_{(k_0)}^T \omega_{(k_0)} = \omega_{(k_0)}^T \omega_{(k_0)} \quad (6)$$

我们将式(6)的值等于0可以得到 $\lambda C_{k_0} = \omega$ .定义 $G_{k_0} = \lambda_{(k_0)}^T \omega_{(k_0)}$ , $C_{k_0}^T G_{k_0} = \left(\lambda C_{k_0}\right)_{(k_0)}^T \omega_{(k_0)} = \omega_{(k_0)}^T \omega_{(k_0)}$ ,因此:

$$\mathbf{G}_{k_0} = \mathbf{C}_{k_0} \mathbf{C}_{k_0}^T \mathbf{C}_{k_0} \quad (7)$$

我们使用 SVD 矩阵分解算法<sup>[11]</sup>分解矩阵  $\mathbf{G}_{k_0}$  得到  $\mathbf{G}_{k_0} = \mathbf{P}_{k_0} \mathbf{S}_{k_0} \mathbf{Q}_{k_0}^T$ , 其中  $\mathbf{P}_{k_0}$  和  $\mathbf{Q}_{k_0}$  是正交矩阵,  $\mathbf{P}_{k_0}$  是左奇异矩阵,  $\mathbf{Q}_{k_0}$  是右奇异矩阵,  $\mathbf{S}_{k_0}$  是对角矩阵. 最后, 我们可以得到

$$\mathbf{C}_{k_0} = \mathbf{P}_{k_0} \mathbf{Q}_{k_0}^T \quad (8)$$

通过以上计算, 我们可以得到  $\partial\varphi/\partial\mathbf{C}_{k_0}$  和  $\mathbf{C}_{k_0}$  的值, 同理可以计算  $\partial\varphi/\partial\mathbf{D}_{k_0}$  和  $\mathbf{D}_{k_0}$  的值. 在此基础上, 我们使用 ALS 算法最小化目标函数  $\varphi$  计算  $\{\mathbf{C}_k\}$  和  $\{\mathbf{D}_k\}$  的值.

在求得  $\{\mathbf{C}_k\}$  和  $\{\mathbf{D}_k\}$  的值之后, 我们还需要解决问题 2, 从而完成张量分解过程.

**问题 2** 给定一个张量  $\chi \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_k}$  和两个因子矩阵集  $\{\mathbf{C}_k\}$  和  $\{\mathbf{D}_k\}$ ,  $\mathbf{C}_k \in \mathbb{R}^{I_k \times H_k}$ ,  $\mathbf{D}_k \in \mathbb{R}^{I_k \times H_k}$ . 我们要得到一个核心张量  $\tilde{\chi}$ ,  $\tilde{\chi} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_k}$ , 最小化目标函数  $\psi$ ,

$$\psi = \left\| \chi \prod_k \times_k \mathbf{C}_k - \tilde{\chi} \prod_k \times_k \mathbf{D}_k \right\|_F \quad (9)$$

同时,  $\{\mathbf{C}_k\}$  和  $\{\mathbf{D}_k\}$  满足条件 1 和条件 2. 求解问题 2 需要使用如下引理:

**引理 2** 给定目标函数  $\delta = \text{tr} \left( \left( \chi \prod_k \times_k \mathbf{C}_k \right) \prod_k \times_k \mathbf{C}_k^T \right)$ , 其中  $\chi \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_k}$ ,  $\mathbf{C}_k \in \mathbb{R}^{I_k \times H_k}$ , 并且  $\mathbf{C}_k$  满足条件 2, 则目标函数  $\delta$  对  $\chi$  的偏导数  $\partial\delta/\partial\chi = \varepsilon \left( \chi \prod_k \times_k \mathbf{C}_k \right) \prod_k \times_k \mathbf{C}_k^T$ , 其中  $\varepsilon$  是一个常数.

**证明** 我们使用  $\kappa$  表示  $\tau \prod_k \times_k \mathbf{C}_k$ , 其中  $\tau$  是单位张量,  $\tau \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_k}$ . 那么目标函数  $\delta$  可以写成  $\delta = \text{tr}(\chi \kappa \kappa^T \chi^T)$ , 使用  $\rho$  表示  $\kappa \kappa^T$ , 通过函数求导规则, 我们可以得到如下等式:  $\partial\delta/\partial\chi = \partial\text{tr}(\chi \rho \chi^T)/\partial\chi = (\rho \chi^T + \rho^T \chi)^T$ , 因此,  $\partial\delta/\partial\chi = 2\chi \kappa \kappa^T = \varepsilon \left( \chi \prod_k \times_k \mathbf{C}_k \right) \prod_k \times_k \mathbf{C}_k^T$ , 证毕.

我们使用  $\eta$  表示  $\chi \prod_k \times_k \mathbf{C}_k$ ,  $\gamma$  表示  $\tau \prod_k \times_k \mathbf{D}_k$ , 其中  $\tau$  是单位张量,  $\tau \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_k}$ , 则式 (8) 可写成  $\psi = \|\eta - \tilde{\chi} \gamma\|_F = \text{tr}((\eta - \tilde{\chi} \gamma)^T (\eta - \tilde{\chi} \gamma))$ . 因此, 函数  $\psi$  对  $\tilde{\chi}$  的偏导数为  $\partial\psi/\partial\tilde{\chi} = (\partial\text{tr}(\eta^T \eta) - 2\partial\text{tr}(\tilde{\chi}^T \eta \gamma^T) + \partial\text{tr}(\tilde{\chi} \gamma \gamma^T \tilde{\chi}^T))/\partial\tilde{\chi}$ . 根据函数求导规则和引理 2 可得:

$$\partial\psi/\partial\tilde{\chi} = \varepsilon(\tilde{\chi} \gamma \gamma^T - \eta \gamma^T) \quad (10)$$

其中  $\varepsilon$  是一个常数. 我们将式 (10) 的值等于 0 可以等到问题 2 的最终解.

$$\tilde{\chi} = \left( \chi \prod_k \times_k \mathbf{C}_k \right) \prod_k \times_k \mathbf{D}_k^T \quad (11)$$

综上所述, TTD 算法共分为两步, 首先通过初始张量  $\chi$ 、模板张量  $\nu$  计算因子矩阵集  $\{\mathbf{C}_k\}$  和  $\{\mathbf{D}_k\}$ , 其次通过初始张量  $\chi$ 、 $\{\mathbf{C}_k\}$  和  $\{\mathbf{D}_k\}$  计算得出  $\tilde{\chi}$ . 具体伪代码如下

算法 1 所示. 算法 1 第 2 行 *threshold* 代表 ALS 算法的迭代次数, 第 4 行和第 9 行的函数 *func* 代表式 (6)、(7) 的计算过程, 第 14 行的函数 *calc* 代表式 (11), 可求得核心张量  $\tilde{\chi}$ .

#### 算法 1 基于模板的张量分解 (TTD 算法)

```

Input: Tensor  $\chi$ , Template tensor  $\nu$ 
Output: The core tensor  $\tilde{\chi}$ , factor matrices  $\{\mathbf{C}_k\}$  and  $\{\mathbf{D}_k\}$ 
1: Initialize factor matrices  $\{\mathbf{C}_k\}$  and  $\{\mathbf{D}_k\}$ 
2: for  $i = 1$  to threshold do
3:   for  $k_0 = 1$  to  $K$  do
4:      $\mathbf{G}_{k_0} = \text{func}(\chi, \nu, \{\mathbf{C}_k\}, \{\mathbf{D}_k\}, k_0)$ ;
5:      $\mathbf{P}_{k_0}, \mathbf{S}_{k_0}, \mathbf{Q}_{k_0} = \text{SVD}(\mathbf{G}_{k_0})$ ;
6:      $\mathbf{C}_{k_0} = \mathbf{P}_{k_0} \mathbf{Q}_{k_0}^T$ ;
7:   end
8:   for  $k_0 = 1$  to  $K$  do
9:      $\mathbf{G}_{k_0} = \text{func}(\chi, \nu, \{\mathbf{D}_k\}, \{\mathbf{C}_k\}, k_0)$ ;
10:     $\mathbf{P}_{k_0}, \mathbf{S}_{k_0}, \mathbf{Q}_{k_0} = \text{SVD}(\mathbf{G}_{k_0})$ ;
11:     $\mathbf{D}_{k_0} = \mathbf{P}_{k_0} \mathbf{Q}_{k_0}^T$ ;
12:  end
13: end
14:  $\tilde{\chi} = \text{calc}(\chi, \{\mathbf{C}_k\}, \{\mathbf{D}_k\})$ 
15: return  $\tilde{\chi}, \{\mathbf{C}_k\}, \{\mathbf{D}_k\}$ 

```

### 3.4 Bi-LSTM 分类模型

如图 2 所示, 在张量分解完成之后, 我们使用 Bi-LSTM 分类模型来完成案件的罪名认定. Bi-LSTM 神经网络的训练过程描述如下:

(1) 将张量分解后的核心张量集  $\{\tilde{\chi}^{(n)}\}$  作为 Bi-LSTM 分类模型的输入, 每个核心张量  $\tilde{\chi}^{(n)}$  是使用 TTD 算法将案件表示张量集  $\{\chi^{(n)}\}$  进行张量分解得到的;

(2) Bi-LSTM 神经网络的输出层  $\{h_d^{(n)}\}$  是 softmax 层, 用于将输出向量映射到罪名类别集  $\{L^{(n)}\}$  上. 我们使用交叉熵作为模型的损失函数, 记为 LossF. 模型如图 3 所示;

(3) Bi-LSTM 神经网络通过前向传播算法和误差反

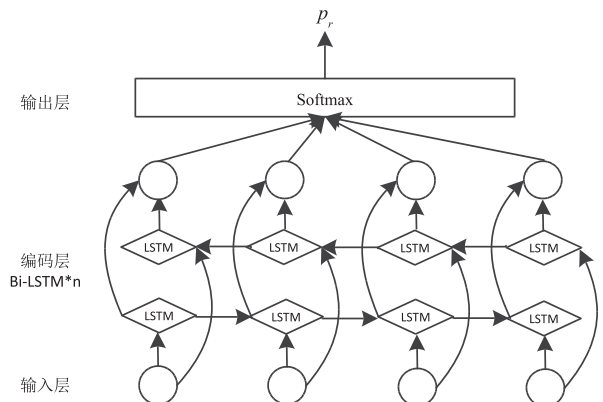


图3 Bi-LSTM分类模型

向传播算法不断更新神经网络的参数值,包括权重矩阵  $\{\mathbf{w}_d\}$  和偏置项  $\{\mathbf{b}_d\}$  和模板张量  $\nu$ ,即需要计算损失函数  $LossF$  对以上参数的偏导数,即  $\partial LossF/\partial \mathbf{w}_d$ 、 $\partial LossF/\partial \mathbf{b}_d$  和  $\partial LossF/\partial \nu$ . 我们使用 Rmsprop 优化算法完成权重矩阵  $\{\mathbf{w}_d\}$  和偏置项  $\{\mathbf{b}_d\}$  和模板张量的训练优化.

在使用反向传播算法训练网络的过程中,除了参数和偏置项,为了优化模板张量  $\nu$ ,我们还需要计算  $\partial LossF/\partial \nu$ ,根据函数求导规则可得:

$$\partial LossF/\partial \nu = \sum_n^N ((\partial LossF/\partial \tilde{\chi}^{(n)}) (\partial \tilde{\chi}^{(n)}/\partial \nu)) \quad (12)$$

通过式 (11) 得  $\partial \tilde{\chi}^{(n)}/\partial \nu = \partial \left[ \left( \chi^{(n)} \prod_k^K \times_k C_k^{(n)} \right) \prod_k^K \times_k D_k^{(n)T} \right] / \partial \nu$ . 由式 (6)、(7) 的推导过程可知,  $\{C_k^{(n)}\}$  和  $\{D_k^{(n)T}\}$  都可以表示为  $\nu$  的函数,通过求导规则计算  $\partial \tilde{\chi}^{(n)}/\partial C_k^{(n)}$ 、 $\partial C_k^{(n)}/\partial \nu$ 、 $\partial \tilde{\chi}^{(n)}/\partial D_k^{(n)T}$  和  $\partial D_k^{(n)T}/\partial \nu$ , 可得:

$$\partial \tilde{\chi}^{(n)}/\partial \nu = \sum_k^K calC(C_k^{(n)}) + \sum_k^K calD(D_k^{(n)T}) \quad (13)$$

其中:

$$calC(C_k^{(n)}) = \chi^{(n)} \prod_{k_1 \neq k}^K \times_{k_1} C_{k_1}^{(n)} \times_k \frac{\partial C_k^{(n)}}{\partial \nu} \prod_{k_2}^K \times_{k_2} D_{k_2}^{(n)T} \quad (14)$$

$$calD(D_{k_2}^{(n)T}) = \chi^{(n)} \prod_{k_1}^K \times_{k_1} C_{k_1}^{(n)} \times_k \frac{\partial D_{k_2}^{(n)T}}{\partial \nu} \prod_{k_2 \neq k}^K \times_{k_2} D_{k_2}^{(n)T} \quad (15)$$

我们将式(7)两侧对  $\nu$  求偏导数可得等式  $\partial G_k^{(n)}/\partial \nu - C_k^{(n)} (\partial G_k^{(n)T}/\partial \nu) C_k^{(n)} = (\partial C_k^{(n)}/\partial \nu) G_k^{(n)T} C_k^{(n)} + C_k^{(n)} G_k^{(n)T} (\partial C_k^{(n)}/\partial \nu)$ , 该等式是 Sylvester 矩阵方程,可使用 hessenberg-schur 算法求解  $\partial C_k^{(n)}/\partial \nu$ , 同理可求得  $\partial D_k^{(n)T}/\partial \nu$ , 从而通过式 (13)、(14) 计算得出  $\partial LossF/\partial \nu$ . 通过  $\partial LossF/\partial \nu$  的值,在反向传播训练过程中,通过 Rmsprop 优化算法可不断优化模板张量. 具体的算法过程如算法 2 所示,第 2 行  $max\_iter$  代表训练迭代的次数,第 4 到第 12 行代表 Bi-LSTM 网络的训练步骤,第 7 行的 TTD 函数代表本文提出的模板张量分解算法,第 11 行说明了我们使用 Rmsprop 算法完成网络参数及模板张量的训练优化. 随着模板张量  $\nu$  不断的优化,TTD 张量分解后的核心张量所包含的特征信息越来越丰富,将其输入到 Bi-LSTM 中能够使损失函数  $LossF$  逐渐收敛并稳定.

案件张量  $\chi^{(n)}$  属于罪名集  $\{L^{(n)}\}$  内各个罪名的预测概率向量由 Bi-LSTM 网络最后的 Softmax 层计算得到,记为  $L_1^{(n)}$ ,其中第  $q$  个元素为  $L_{1q}^{(n)}$ ,其损失函数计算公式为:

$$LossF(\{L^{(n)}\}, \{L_1^{(n)}\}) = \sum_n^N \sum_q^Q L_q^{(n)} \log L_{1q}^{(n)} \quad (16)$$

其中  $N$  代表案件样本数量,  $Q$  代表罪名类别数量,  $L_q^{(n)}$  是类别指示变量,如果该类别和样本的类别相同就是 1, 否则是 0.

## 算法 2 模板张量的训练优化算法

---

Input:  $\{\chi^{(n)}, L^{(n)}\}$ , the size of  $\nu$ ,  $\{w_f\}$ ,  $\{w_i\}$ ,  $\{w_o\}$ ,  $\{w_c\}$ ,  $\{b_f\}$ ,  $\{b_i\}$ ,  $\{b_o\}$  and  $\{b_c\}$

Output: Template tensor  $\nu$ , parameters of Bi-LSTM  $\{w_f\}$ ,  $\{w_i\}$ ,  $\{w_o\}$ ,  $\{w_c\}$ ,  $\{b_f\}$ ,  $\{b_i\}$ ,  $\{b_o\}$  and  $\{b_c\}$

---

- 1: Initialize template tensor  $\nu$ , parameters of Bi-LSTM  $\{w_f\}$ ,  $\{w_i\}$ ,  $\{w_o\}$ ,  $\{w_c\}$ ,  $\{b_f\}$ ,  $\{b_i\}$ ,  $\{b_o\}$  and  $\{b_c\}$
- 2: for  $iter = 1$  to  $max\_iter$  do
- 3: Set  $batch\_indices$  to 0;
- 4: while  $batch\_indices \leq dataset\_size$  do
- 5: Set  $\partial LossF/\partial \nu$ ,  $\{\partial LossF/\partial b_f\}$ ,  $\{\partial LossF/\partial w_i\}$ ,  $\{\partial LossF/\partial w_o\}$ ,  $\{\partial LossF/\partial w_c\}$ ,  $\{\partial LossF/\partial b_i\}$ ,  $\{\partial LossF/\partial b_o\}$ ,  $\{\partial LossF/\partial b_c\}$  to 0;
- 6: for  $i = batch\_indices$  to  $(batch\_indices + batch\_size)$  do;
- 7:  $\tilde{\chi}^{(i)}, \{C_k\}, \{D_k\} = TTD(\chi^{(i)}, \nu)$ ;
- 8:  $L_1^{(i)} = Bi\_LSTM\_forwardprop(\tilde{\chi}^{(i)}, \{w_f\}, \{w_i\}, \{w_o\}, \{w_c\}, \{b_f\}, \{b_i\}, \{b_o\}, \{b_c\})$ ;
- 9:  $\partial LossF/\partial \nu$ ,  $\{\partial LossF/\partial b_f\}$ ,  $\{\partial LossF/\partial w_i\}$ ,  $\{\partial LossF/\partial w_o\}$ ,  $\{\partial LossF/\partial w_c\}$ ,  $\{\partial LossF/\partial b_i\}$ ,  $\{\partial LossF/\partial b_o\}$ ,  $\{\partial LossF/\partial b_c\} + Bi\_LSTM\_backprop(\tilde{\chi}^{(i)}, L_1^{(i)}, L^{(i)}, \{w_f\}, \{w_i\}, \{w_o\}, \{w_c\}, \{b_f\}, \{b_i\}, \{b_o\}, \{b_c\})$ ;
- 10: end
- 11:  $\nu$ ,  $\{w_f\}$ ,  $\{w_i\}$ ,  $\{w_o\}$ ,  $\{w_c\}$ ,  $\{b_f\}$ ,  $\{b_i\}$ ,  $\{b_o\}$ ,  $\{b_c\} = Rmsprop(\nu, \{w_f\}, \{w_i\}, \{w_o\}, \{w_c\}, \{b_f\}, \{b_i\}, \{b_o\}, \{b_c\}, \partial LossF/\partial \nu, \{\partial LossF/\partial b_f\}, \{\partial LossF/\partial w_i\}, \{\partial LossF/\partial w_o\}, \{\partial LossF/\partial w_c\}, \{\partial LossF/\partial b_i\}, \{\partial LossF/\partial b_o\}, \{\partial LossF/\partial b_c\})$ ;
- 12: end
- 13: end
- 14: return  $\nu$ ,  $\{w_f\}$ ,  $\{w_i\}$ ,  $\{w_o\}$ ,  $\{w_c\}$ ,  $\{b_f\}$ ,  $\{b_i\}$ ,  $\{b_o\}$  and  $\{b_c\}$

---

## 4 实验与结果分析

### 4.1 实验数据集

实验数据来自最高人民检察院检察技术信息研究中心提供的 2015 ~ 2017 年共 20000 余份已脱敏的真实案件,覆盖了盗窃、故意伤害等 200 余个刑事罪名. 考虑到本文提出的方法主要用于相似罪名的精准识别,如表 1 所示,我们从中选择了司法实务中容易混淆并且是比较难以区分的三组 7 个罪名的案件进行实验. 每个罪名选取 500 件案件,形成了研究工作所用的案件数据集.

表 1 实验数据集的罪名

序号	罪名
第一组	抢劫罪
	抢夺罪
	敲诈勒索罪
第二组	非法吸收公众存款罪
	集资诈骗罪
第三组	诈骗罪
	招摇撞骗罪

## 4.2 数据预处理

实验数据预处理主要包括三个步骤:

(1) 将案件卷宗的文本内容划分为四个部分: 案件的陈述、侦查机关认定的犯罪事实以及检察机关的公诉意见, 这四部分内容分布在起诉意见书、公诉案件审查报告等文件内, 我们将分布在多个文件内的内容进行合并和精炼;

(2) 提取出以上四部分内容后, 进行过滤和清洗, 去掉文本中的冗余词、停用词以及语气助词等没有意义的内容;

(3) 使用大型中文语料库和 word2vec 工具将文本中的词转为 200 维度的词向量, 从而每部分内容可以形成一个矩阵, 合并四个矩阵即可形成表征案件的三维张量。

经过以上三个步骤, 生成了案件定罪方法的输入张量, 考虑到分类算法需要张量的大小一致, 而 95% 的案件一项内容的词汇量小于 300, 因此, 如果案件词汇量小于 300, 我们使用 padding 操作保证张量大小的一致; 对于词汇量大于 300 的案件, 我们使用词汇的 TF-IDF 权重来选择词汇。由此可知, 数据集内任意一个案件的表示张量  $\chi \in \mathbb{R}^{300 \times 200 \times 4}$ 。

## 4.3 实验设计

为了验证提出的司法案件定罪方法以及该方法在司法案件定罪方面的性能, 我们进行了一系列对比实验, 包括:

(1) 从分析模板张量形状 (Shape) 的最优大小的角度, 比较不同大小的模板张量对于分类性能的影响, 综合考虑训练时间和分类性能, 找到最优模板张量的形状;

(2) 从分析张量分解算法性能的角度, 将常用的 Tucker 和 CP 张量分解算法与本文提出的 TTD 张量分解算法进行比较;

(3) 从分析深度神经网络性能的角度, 我们选择了 TextCNN<sup>[12]</sup>、TextRNN<sup>[13]</sup>、LSTM、GRU<sup>[14]</sup> 和 Bi-GRU<sup>[15]</sup> 五种常用于文本分类的深度神经网络模型与本文选择的 Bi-LSTM 进行比较。

实验选择 TensorFlow 工具训练以上提到的所有神经网络模型, 超参数设定为隐藏层数为 3, 隐藏层节点数为 512, 学习率为 0.001。我们将 7 个罪名的案件共 3500 个案件, 分别按照 4:1 的比例将数据集分成训练集和测试集。使用训练集数据训练神经网络模型, 最后使用测试集测试各个模型的性能。模板张量的初始化是影响 Bi-LSTM 神经网络训练收敛速度和分类准确率的关键因素之一, 考虑到张量分解的性质以及实验结果表明, 模板张量的秩越低, 张量内各行列线性相关性越强, 则我们分类算法的准确率越低。因此, 模板张量应初始化为一个高秩张量。

## 4.4 实验结果分析

首先, 我们将模板张量设置为不同的形状, 并使用本文提出的算法完成案件分类任务。表 2 展示了不同形状的模板张量的分类算法准确率。从表中结果可以看到, 随着模板张量行数、列数和深度三个维度的增长, 准确率也逐步提高, 但提高幅度也随之下降。可见, 模板张量的形状越大, TTD 张量分解后得到的核心张量的形状也越大, 包含案件的关键信息也越多, 从而案件分类的准确率也越高。而提高幅度的逐渐下降, 考虑是因为数据集内的一些疑难案件并不会随着输入信息的提升而准确分类。但是模板张量的形状越大, 需要训练的参数也越多, 同时张量分解计算所需要的时间也越长。综合考虑时间和准确率两个方面, 我们选择 (100, 100, 3) 大小的模板张量进行接下来两项实验。

表 2 不同模板张量大小的案件分类准确率

模板张量大小	准确率	参数量
(50, 50, 2)	0.91	5000
(50, 50, 3)	0.92	7500
(100, 100, 2)	0.95	20000
(100, 100, 3)	0.96	30000
(150, 100, 3)	0.97	45000
(150, 150, 3)	0.97	67500

其次, 我们选择不同的深度神经网络模型与张量分解算法组合来完成案件分类任务, 表 3 和表 4 展示了不同案件分类算法的准确率和 F1 值。从表中结果可以看到, Bi-LSTM 结合 TTD 张量分解算法取得了所有比较算法中最好的分类效果, TextCNN 在所有比较算法中效果较差, 可见, 卷积神经网络在自然语言处理任务上的效果没有循环神经网络的效果好, 毕竟案件文本作为序列数据, 使用循环神经网络模型处理更加适合。而在其他循环神经网络模型中, Bi-LSTM 和 Bi-GRU 由于兼顾上下文信息, 取得了较好的分类效果, 准确率和 F1 值相较于其他算法明显要好。Bi-GRU 是 Bi-LSTM 网络的一种变体, Bi-LSTM 使用输出门控制网络神经单元的

输出,而 Bi-GRU 直接将输出传给下一个神经单元. 不过从使用 TTD 优化的效果来看,TTD 结合 Bi-LSTM 后准确率提升的更高,从不使用张量分解的 0.83 提高到

0.96. 分析原因,我们认为 GRU 相比 LSTM 来说具有更高的集成度和较少的可调参数,所以 TTD 算法对其的优化效果没有 Bi-LSTM 好.

表 3 分类算法准确率

	TextCNN	TextRNN	LSTM	Bi-LSTM	GRU	Bi-GRU
无	0.76	0.79	0.81	0.83	0.81	<b>0.85</b>
CP 分解	0.83	0.87	0.89	<b>0.92</b>	0.90	0.91
Tucker 分解	0.84	0.85	0.87	0.90	0.89	<b>0.92</b>
TTD 分解	<b>0.89</b>	<b>0.91</b>	<b>0.93</b>	<b>0.96</b>	<b>0.92</b>	<b>0.95</b>

表 4 分类算法 F1 值

	TextCNN	TextRNN	LSTM	Bi-LSTM	GRU	Bi-GRU
无	0.64	0.68	0.69	0.74	0.71	<b>0.77</b>
CP 分解	0.72	0.80	0.82	<b>0.87</b>	0.83	0.84
Tucker 分解	0.75	0.77	0.80	0.84	0.82	<b>0.86</b>
TTD 分解	<b>0.83</b>	<b>0.84</b>	<b>0.88</b>	<b>0.92</b>	<b>0.87</b>	<b>0.90</b>

图 4 比较了 LSTM 与 Bi-LSTM 结合张量分解算法后的收敛速度和准确率. 从实验结果可分析出,在深度

神经网络分类模型之前加上张量分解,可以有效提升分类算法的收敛速度和准确性. 这是因为张量分解可以有效去除初始张量内的冗余信息,同时保留初始张量元素之间的内在联系,从而有效提升之后的分类算法准确性;同时,张量分解算法降低了分类算法同时,张量分解算法降低了分类算法输入张量的维度,有效提升了分类算法的收敛速度. 通过三种张量分解算法比较可以发现,Tucker 与 CP 张量分解算法在分类模型的优化效果方面差距很小,这是因为 CP 分解是 Tucker 分解的一种特殊形式,都是一种高阶奇异值分解算法. 同时可发现本文提出的 TTD 算法结合分类模型后取得了更好的分类效果,这是因为 TTD 算法引入了模板张量,模板张量经过深度神经网络分类模型的训练过程不断的优化,使得模板张量具有更适用于案件分类的张量结构和特征信息,而 TTD 分解过程就是使分解后的核心张量与模板张量越来越近似的过程,因此当分解后的核心张量输入到深度神经网络分类模型,会取得更好的分类效果.

## 5 结论

本文提出了一种用于司法案件定罪的智能方法. 该方法结合了基于模板的张量分解算法和 Bi-LSTM 神经网络分类模型. 相比于传统的张量分解算法,本文提出的基于模板的张量分解算法在分解过程中引入了模板张量,通过分类模型不断优化模板张量,从而使得分解得到的核心张量具有更适用于案件分类的张量结构和特征信息,从而有效提升司法案件智能定罪的准确率. 在与其他方法的对比实验中,本文提出的方法表现出了较好的性能. 下一步工作主要包含两个方面:一方

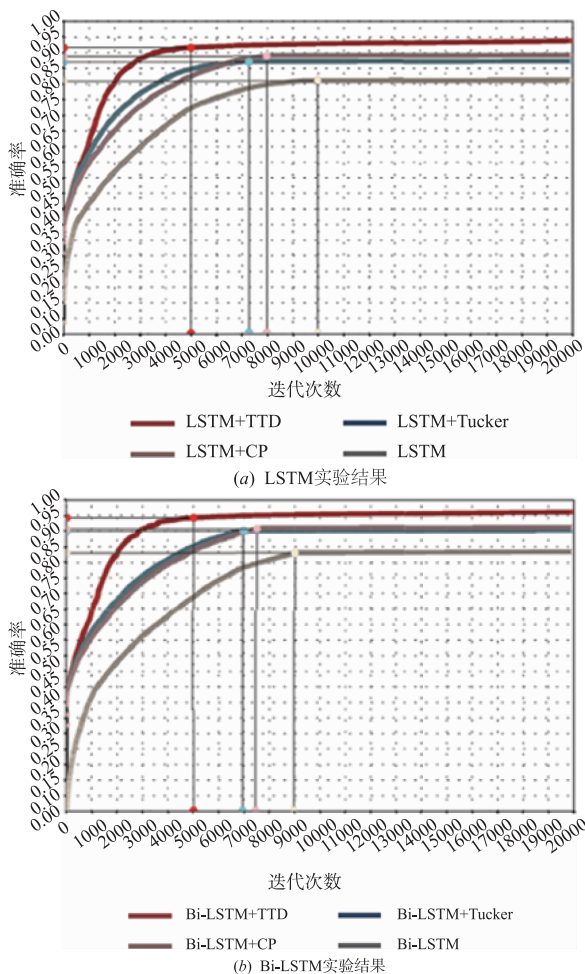


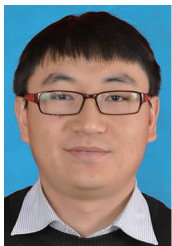
图4 LSTM与Bi-LSTM分类算法的准确率实验结果

面需要进一步提升该方法在案件定罪方面的准确率,如引入中文语言预训练模型替换 word2vec;另一方面,我们会考虑将该方法扩展到如刑期预测等其他智能辅助办案场景中。

#### 参考文献

- [1] 黄俏娟,罗旭东. 人工智能与法律结合的现状与发展趋势[J]. 计算机科学,2018,45(12):1-11.  
Huang Qiao-juan, Luo Xu-dong. State-of-the-art and development trend of artificial intelligence combined with law [J]. Computer Science, 2018, 45(12):1-11. (in Chinese)
- [2] Octavia-Maria S, Marcos Z, Shervin M, Mihaela V, Liviu P D, Josef V G. Exploring the use of text classification in the legal domain[A]. Proceedings of the Second Workshop on Automated Semantic Analysis of Information in Legal Texts[C]. London, UK: ACM, 2017. 22-27.
- [3] Octavia-Maria S, Marcos Z, Mihaela V, Josef V G. Predicting the law area and decisions of French supreme court cases[A]. Proceedings of the International Conference Recent Advances in Natural Language Processing[C]. Varna, Bulgaria: ACM, 2017. 56-63.
- [4] 林志宏,池宏,许保光. 基于卷积神经网络的公安案件文本语义特征提取方法研究[J]. 数学的实践与认识, 2017,46(17):129-142.  
Lin Zhi-hong, Chi Hong, Xu Bao-guang. Research of criminal case semantic feature extraction method based on the convolutional neural network[J]. Mathematics in Practice and Theory, 2017, 46(17):129-142. (in Chinese)
- [5] Xiao G Y, Mo J Q, Even C, Chen H. Multi-task CNN for classification of Chinese legal questions[A]. IEEE 14th International Conference on e-Business Engineering [C]. Shanghai, China: IEEE, 2017. 126-132.
- [6] Guo X D, Zhang H L, Ye L, Li S. Learning users' intention of legal consultation through pattern-oriented tensor decomposition with Bi-LSTM[J]. Wireless Communications and Mobile Computing, 2019, 32(3):1-16.
- [7] Kolda T G, Bader B W. Tensor decompositions and applications[J]. SIAM Review, 2009, 51(3):455-500.
- [8] Chen T, Xu R F, He Y L, Wang X. Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN[J]. Expert Systems with Applications, 2017, 72(4):221-230.
- [9] Tucker LR. Some mathematical notes on three-mode factor analysis[J]. Psychometrika, 1966, 31(3):279-311.
- [10] Harshman RA. Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multimodal factor analysis[J]. UCLA Working Papers in Phonetics, 1970, 45(16):1-84.
- [11] Vozalis M G, Margaritis K G. Using SVD and demographic data for the enhancement of generalized collaborative filtering[J]. Information Sciences, 2007, 177(15):3017-3037.
- [12] Kim Y. Convolutional neural networks for sentence classification[A]. Proceedings of the 2014 Conference on empirical methods in natural language processing[C]. Doha, Qatar: ACL, 2014. 1746-1751.
- [13] Donahue J, Hendricks L A, Guadarrama S. Long-term recurrent convolutional networks for visual recognition and description[A]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition [C]. Boston, USA: IEEE, 2015. 2625-2634.
- [14] Cho K, Bart V M, Caglar G, Fethi M. Learning phrase representations using RNN encoder-decoder for statistical machine translation[A]. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing[C]. Doha, Qatar: ACL, 2014. 1724-1734.
- [15] Donahue J, Hendricks L A, Guadarrama S. Ask the GRU: Multitask learning for deep text recommendations[A]. Proceedings of the 10th ACM Conference on Recommender Systems[C]. Como, Italy: ACM, 2016. 107-114.

#### 作者简介



李大鹏(通信作者) 男,1982年1月出生,辽宁沈阳人,东北大学博士研究生,主要研究方向为人工智能、自然语言理解等。  
E-mail: lidp@neusoft.com



陈剑 男,1980年9月出生,东北大学副教授,硕士生导师,主要研究方向为人工智能、网络与信息安全。  
E-mail: chen.jian@neusoft.com