

文章编号: 2095-2163(2023)04-0091-06

中图分类号: TP391

文献标志码: A

# 基于 AlBert-Tiny-DPCNN 的案件事实倾向性类别预测

施君可

(浙江理工大学 信息学院, 杭州 310018)

**摘要:** 近年来,随着智慧司法的推进,中国裁判文书作为重要的研究对象,衍生了诸多任务,但针对裁判文书的研究大多基于刑事案件,缺乏对民事案件领域下的研究。本文结合预训练词向量、文本分类模型等技术,对民间借贷这一细分领域下的案件事实标签预测进行研究,为现有的案件事实提供同类别的有参考价值的裁判文书,减少相关工作者在大量数据中寻找所耗费的时间。本文提出了基于 AlBert-Tiny-DPCNN 的分类模型,该模型采用注意力机制与标签平滑归一化技术来提高模型的精度,并在实验数据集上验证了模型的有效性。

**关键词:** 深度学习; 裁判文书; 文本分类; 预训练词向量

## Chinese judicial document processing by utilizing case fact label classification based on AlBert-Tiny-DPCNN

SHI Junke

(School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China)

**[Abstract]** In recent years, with the advancement of intelligent justice, judicial documents in China, as an important research object, have derived many tasks. However, most of the research on judicial documents is based on criminal cases, and there is a lack of research on civil cases. Combining pre-trained word vectors, text classification models, and other technologies, this paper studies the prediction of case facts in the subdivision field of private lending so as to provide the existing case facts with similar judgment documents with reference value and reduce the time spent by relevant workers looking for them in a large amount of data. In this paper, a classification model based on AlBert-Tiny-DPCNN is proposed, attention mechanisms and label smoothing regularization techniques are used to improve the accuracy of the model, and the validity of the model is verified on experimental data sets.

**[Key words]** deep learning; Chinese judicial document; text classification; pre-trained word vector

## 0 引言

近年来,裁判文书结合 NLP 衍生出诸多任务,如:类案检索、信息抽取等,但大多研究均基于刑事案件,缺乏对民事案件领域下的研究。而且,由于民事案件的判决结果不会出现原告“胜诉、败诉”等确切字句,只有“支持原告部分请求”、“驳回原告其余请求”等字样,只能对裁判文书的案件事实进行了倾向性分类。此外,民间借贷案件存在借贷金额不同、借贷关系复杂等因素,可能还存在“同案不同判”的问题<sup>[1]</sup>,并且相关从事人员不易从大量的文书中寻找到同类别的参考文书。针对这些状况,本文基于民间借贷一审判判决书数据集,以案件事实作为输入,通过深度学习技术进行案件事实倾向性类

别预测的研究。本文主要工作如下:

(1)提出了基于 AlBert-Tiny-DPCNN 的案件事实标签分类模型。该模型使用 AlBert-Tiny 进行词嵌入,使用 DPCNN 的特征提取结构捕获文本语义,并使用带类别权重的焦点损失进行模型训练。实验结果表明,相比于对照模型,该模型的准确率与加权  $F_1$  值更高,准确率能够达到 79.65%。

(2)本文在上述模型的基础上,采用注意力机制增强特征提取层输出向量的重要特征、减弱无用特征,使用标签平滑归一化方法提升模型的泛化能力,准确率提升到了 81.22%,增加了 1.97%。

## 1 研究现状

本文研究了近几年来裁判文书结合深度学习的

作者简介: 施君可(1997-),男,硕士研究生,主要研究方向:文本分类。

通讯作者: 施君可 Email: 15858130053@163.com

收稿日期: 2022-05-20

哈尔滨工业大学主办 ◆ 学术研究与应用

分类模型现状。王文广等学者<sup>[2]</sup>基于 HAN 和 DPCNN 提出了法律判决预测模型 HAC, 在刑期预测等各项判决预测任务中表现良好。王业沛等学者<sup>[3]</sup>探索了不同层数的 LSTM 模型在预测判决结果的倾向性分类时的效果, 发现 3 层 LSTM 在实验中准确率较高, 但仅使用 LSTM 很难凸显文本的重要信息。王宁等学者<sup>[4]</sup>使用基于注意力的 BiGRU 模型来预测判决结果的倾向性, 实验发现模型能在一定程度上进行有效预测, 但模型的准确率仍有提升空间。自 2018 年底始, 预训练模型从最开始的 Word2Vec、Glove 过渡到了 BERT 系列、ERNIE、XLNet、MPNet 等模型<sup>[5-6]</sup>。目前, 预训练语言模型被广泛使用在文本分类任务中。王立梅等学者<sup>[7]</sup>对比了 TextCNN、TextRNN、Transformer、Bert 模型, 然后优化了最优模型的嵌入方式。孟令慈<sup>[8]</sup>基于 Bert-LSTM 模型对刑事案件进行结果分类研究, 通过对序列进行编码和特征融合来获取类别信息。

## 2 模型架构

本文判决结果倾向性分类模型的主要架构如图 1 所示, 主要包括 3 个部分: 嵌入层的作用是将输入文本转为词向量矩阵, 本文比较了 Word2Vec<sup>[9]</sup> 与 Albert-Tiny<sup>[10]</sup> 在本文任务中的效果; 特征提取层是为了提取词向量矩阵中的语义特征, 本文比较了 TextCNN<sup>[11]</sup>、TextRCNN<sup>[12]</sup>、DPCNN<sup>[13]</sup> 三种网络结构对模型的影响; 分类层将输入的神经元数量压缩

至类别个数。此外, 本文对比了交叉熵损失函数和焦点损失函数对模型的影响。

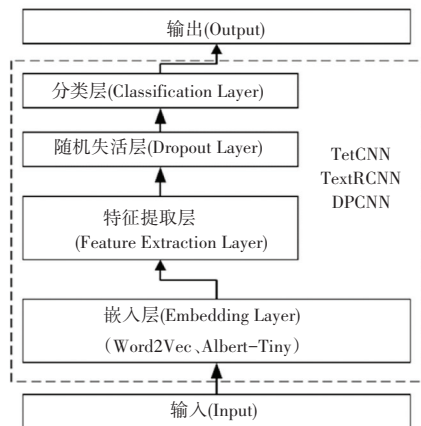


图 1 模型基本框架

Fig. 1 Basic structure of the model

### 2.1 数据采集与处理

本文采用 selenium 工具从中国裁判文书官网 (<https://wenshu.court.gov.cn/>) 爬取民间借贷一审判决书, 并以此构建数据集。裁判文书的文本格式较为规范, 本文通过分析裁判文书的结构, 使用正则提取数据中的案件事实 (fact)、判决理由 (reason)、判决依据 (legal)、判决结果 (verdict) 四个部分。然后, 本文构建了判决结果倾向性类别标签集合, 并结合 reason、legal、verdict 的内容进行数据标注, 将 verdict 转为数值表示。标签集合见表 1。

表 1 类别标签集合

Tab. 1 Collection of category tags

判决结果倾向性分类	类别
对于原告的诉讼请求法院予以驳回; 本案受理费由原告承担。	0
法院对于原告的诉讼请求全部支持; 本案受理费由原告承担。	1
法院对于原告的诉讼请求全部支持; 本案受理费由被告承担。	2
法院对于原告的诉讼请求全部支持; 本案受理费由原告、被告双方承担。	3
法院对于原告的诉讼请求部分支持; 本案受理费由原告承担。	4
法院对于原告的诉讼请求部分支持; 本案受理费由被告承担。	5
法院对于原告的诉讼请求部分支持; 本案受理费由原告、被告双方承担。	6

### 2.2 嵌入层

本文采用 Word2Vec 词向量和 Albert 词向量作为词嵌入进行实验研究。由于文本信息不能直接作为输入, 在词嵌入操作前需要将文本序列化。文本序列化, 即把词序列表示成一系列能够表达文本语义的数字表示, 每个数字代表一个词在词表中的索引。之后, 形状为  $[L]$  的文本序列  $\mathbf{W}$  经过词嵌

入层后会转换成形状为  $[L, D]$  的词嵌入矩阵  $\mathbf{E}$ ,  $L$  为文本序列长度,  $D$  为词向量维度,  $w_i$  表示输入文本序列中的第  $i$  个词在词表中的索引,  $e(w_i)$  表示词表中索引为  $w_i$  的词对应的词向量, 如式 (1) 所示:

$$\mathbf{W} = [w_1, w_2, w_3, \dots, w_L]$$

$$\mathbf{E} = [e(w_1), e(w_2), e(w_3), \dots, e(w_L)] \quad (1)$$

基于此, 进一步展开研究分述如下。

(1) Word2Vec。Word2Vec<sup>[9]</sup>把所有词语投影到  $K$  维的向量空间,将文本内容的处理简化为  $K$  维向量空间中的向量运算。本文使用 jieba 工具对案件事实进行分词,并去除了无关字符与常用停用词。其次,本文使用分词去停后的案件事实构建语料库,使用 gensim 开源工具 Word2Vec 的 skip-gram 算法训练 W2V 词向量表。接着,将分词去停后的案件事实词序列进行文本序列化处理<sup>[4]</sup>,并统一文本长度为 150。最后,本文用文本序列和 W2V 词向量表实现静态词嵌入。

(2) AlBert-Tiny。本文使用了 AlBert-Tiny 进行下游任务的微调,为保证输入文本相同,模型的输入为分词去停后的案件事实。首先,本文使用继承了 transformers 的 BertTokenizer 的 Tokenizer 实现文本序列化。接着,统一文本序列化长度为 300,对超出 298 的数据进行截断,在文本序列首部添加类别符号 '[CLS]',在尾部添加结束符号 '[SEP]',并对不足的文本序列进行填充。最后,使用 transformers 的 AlbertModel 进行下游任务微调,得到需要的 AlbertTiny 词向量。

### 2.3 特征提取层

目前,文本分类模型多种多样,本文主要探索了 TextCNN<sup>[11]</sup>、TextRCNN<sup>[12]</sup>、DPCNN<sup>[13]</sup> 三种特征提取结构在模型中的表现效果。其中,TextCNN 使用不同大小的卷积核来捕获输入文本对应的词向量矩阵中相邻词汇之间的特征信息。因卷积核尺寸通常不会很大,使得 TextCNN 无法捕获长距离信息,故该模型较为适合中短文本场景,并不适合长文本。TextRCNN 使用双向长短期记忆网络获取输入文本的上下文信息,并通过最大池化筛选出最重要的特征,能够获取长距离依赖。DPCNN 通过增加感受野来缓解长距离依赖问题,主要用区域卷积增加模型可编码的输入文本长度,并通过残差结构增加网络深度来缓解梯度弥散问题。

### 2.4 分类层

分类层采用全连接网络结构来提取特征间的关系,该层的输入为特征提取层后的文本语义向量,输出为每个类别的概率,如图 2 所示。图 2 中,第一层全连接的维度形状为  $1 * n$ ,第二层全连接的维度形状为  $1 * (n/2)$ ,输出层(即最终结果输出的网络层)形状为  $1 * C$  的分类结果,这里  $C$  为类别标签的数量,该输出向量通过 softmax 分类器后得到的可能性最高的类别即为预测的类别。除输出层外的 2 个全连接层采用了 LeakRelu 激活函数。另外,可根

据训练效果来决定是否对全连接的输出结果进行批归一化处理。

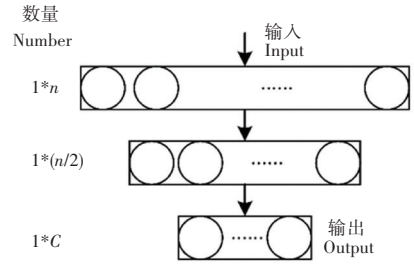


图 2 分类层结构

Fig. 2 Classification layer structure

### 2.5 损失函数

本文的标签分类任务属于多分类问题,故模型的最后一层应使用 softmax 函数进行归一化,并在计算真实标签与预测结果之间的损失值时以 softmax 结果中概率最高的类别为预测标签。本文对比了 Pytorch 框架中的交叉熵损失 (CrossEntropyLoss) 函数以及自定义的焦点损失函数 (FocalLoss) 在模型训练中的表现效果,并在训练过程中增加类别权重  $w$  来缓解类别不平衡问题。由于上述的损失函数内部已使用 softmax 分类器,故本节模型的最后一层不必再使用 softmax 进行归一化。这里,给出带类别权重的交叉熵损失函数见式(2):

$$L_{CE} = \frac{1}{N} \sum_{n=1}^N L_{CE,n}, L_{CE,n}(X_n, Y_n) = - \sum_{i=1}^C w_i y_{n,i} \log \frac{\exp(x_{n,i})}{\sum_{c=1}^C \exp(x_{n,c})} \quad (2)$$

其中,  $N$  是批大小;  $X_n$  是批中第  $n$  条数据的预测标签值,  $x_{n,i}$  为第  $n$  条数据的第  $i$  类别的预测概率;  $Y_n$  是批中第  $n$  条数据的真实标签,  $y_{n,i}$  为第  $i$  类别的真实标签值,只有一个类别为 1,其余类别为 0;  $C$  是类别数目;  $L_{CE}$  为  $N$  条数据的交叉熵损失均值;  $w_i$  是第  $i$  类别的类别权重。接下来,给出焦点损失函数<sup>[14]</sup>见式(3):

$$L_F = \frac{1}{N} \sum_{n=1}^N L_{F,n}, L_{F,n}(p_n) = - \alpha_i (1 - p_n)^\gamma \log p_n$$

$$p_n(X_n, Y_n) = y_{n,i} \left( \frac{\exp(x_{n,i})}{\sum_{c=1}^C \exp(x_{n,c})} \right) + (1 - y_{n,i}) \left( 1 - \frac{\exp(x_{n,i})}{\sum_{c=1}^C \exp(x_{n,c})} \right) \quad (3)$$

其中,  $N$ 、 $X_n$ 、 $x_{n,i}$ 、 $Y_n$ 、 $y_{n,i}$ 、 $C$  的含义与式(2)相同;  $L_F$  为  $N$  条数据的焦点损失均值;  $p_n(X_n, Y_n)$  表示第  $n$  条数据预测的各类别输出为 1 的概率;  $\alpha_i$  是

权重因子;  $\gamma \in [0, 5]$  是聚焦参数。本文将超参  $\alpha_i$  替换为类别权重数组来缓解类别不均衡问题。

### 3 实验与分析

#### 3.1 实验环境

本文使用的操作系统环境为 Ubuntu18.04, 硬件环境为 Intel Xeon Platinum 8163 (Skylake) 2.5 GHz, 内存为 92 GB, GPU 为 1 \* Tesla V100 NVLink-32 G, python 集成平台 Anaconda3, 采用的 Deep Learning 框架为 pytorch, 使用的依赖有: numpy、pandas、regex、jieba、gensim、scikit - learn、pytorch、transformers、matplotlib、tensorboard。

#### 3.2 实验数据与参数设置

本文通过爬虫在中国裁判文书网收集相关的民

事一审判决书, 使用数据筛选过后留下 30 000 条数据来进行实验研究, 并按 6 : 2 : 2 的比例划分训练集、验证集、测试集。

本文的数据集共有 7 个类别, 但各类别的样本数量分布并不均匀, 分别为 {600, 100, 11 300, 480, 70, 15 950, 1 500}, 故对于该问题, 本文在模型训练过程中使用了带类别权重的损失函数, 类别权重  $W$  的计算方式具体如下:

$$W = [w_1, w_2, \dots, w_C] \quad w_i = 1/\log(1.02 + t_i) \quad (4)$$

其中,  $w_i$  为第  $i$  个类别的权重值;  $C$  为类别数;  $t_i$  为第  $i$  个类别样本数占总样本数的比例。为保持对比实验的参数一致性, 本文在模型训练时的参数配置详见表 2。

表 2 对比模型实验中的参数表

Tab. 2 Parameters in the comparative model experiment

参数名	参数值	参数名	参数值
词向量尺寸(embedding size)	256	优化器(optimizer)	AdamW
单次批量(batch size)	512	失活系数(dropout)	0.5
标签平滑参数(label smoothing)	0	学习率(learning rate)	1e-4

#### 3.3 评价指标

本文采用准确率 (*accuracy*)、加权  $F_1$  值 (*Weighted - F<sub>1</sub>*) 作为结果倾向性预测模型的评价指标。准确率计算方法公式具体如下:

$$accuracy = N_{right} / (N_{right} + N_{wrong}) \quad (5)$$

其中,  $N_{right}, N_{wrong}$  分别表示预测正确、错误的数量。

加权  $F_1$  值可以用作多分类问题的评价指标之一, 可以减轻数据不平衡带来的影响。加权  $F_1$  值考虑了样本不均衡的原因, 在计算查准率 (*Precision<sub>weighted</sub>*) 和召回率 (*Recall<sub>weighted</sub>*) 时, 需要各个类别的查准率 (*precision*) 和召回率 (*recall*) 乘以该类在总样本中的占比来求和。而加权  $F_1$  值为 *Precision<sub>weighted</sub>* 和 *Recall<sub>weighted</sub>* 的调和平均数, 详见式 (6)~式 (10):

$$Precision_i = \frac{TP_i}{TP_i + FP_i} \quad (6)$$

$$Precision_{weighted} = \frac{\sum_{i=1}^L (Precision_i * w_i)}{|L|} \quad (7)$$

$$Recall_i = \frac{TP_i}{TP_i + FN_i} \quad (8)$$

$$Recall_{weighted} = \frac{\sum_{i=1}^L (Recall_i * w_i)}{|L|} \quad (9)$$

$$F_{1weighted} = 2 * \frac{Precision_{weighted} * Recall_{weighted}}{Precision_{weighted} + Recall_{weighted}} \quad (10)$$

其中,  $TP_i$  表示真实类别为第  $i$  类、且被模型判定为第  $i$  类的样本数;  $FP_i$  表示真实类别非第  $i$  类、且被模型判定为第  $i$  类的样本数;  $FN_i$  表示真实类别为第  $i$  类、且被模型判定为非第  $i$  类的样本数;  $|L|$  表示类别数量;  $w_i$  表示第  $i$  类别在总样本中的占比。

#### 3.4 对照实验

本文比较了不同的词嵌入方式、特征提取方式、损失函数相互组合的模型在训练时的表现效果, 结果见表 3。表 3 中, W2V 表示 Word2Vec 嵌入方式, TextCNN、TextRCNN、DPCNN 为特征提取方式, CEL 表示交叉熵损失函数、FL 表示焦点损失函数。

从表 3 的实验数据可知, 与交叉熵损失函数相比, 焦点损失函数能够在一定程度上提高模型准确率。相比于 Word2Vec 静态词向量, ALBERT-Tiny 词向量的嵌入效果更好, 能更好地学习词之间的关系, 使得特征提取层输出的特征向量包含更多的文本语义信息。特征提取层使用的网络结构中, DPCNN 的效果相对较好, 能更准确地捕获文本特征。

表 3 对比模型的准确率与加权  $F_1$  值

Tab. 3 Comparing the accuracy and the weighted- $F_1$  value of the model

模型	CEL		FL	
	accuracy	weighted - $F_1$	accuracy	weighted - $F_1$
W2V+TextCNN (baseline)	0.714 3	0.716 6	0.749 5	0.730 7
W2V+TextRCNN	0.635 5	0.645 1	0.697 8	0.640 5
W2V+DPCNN	0.741 9	0.745 3	0.775 4	0.757 5
ALbert-Tiny+TextCNN	0.735 8	0.701 5	0.792 0	0.768 3
ALbert-Tiny+TextRCNN	0.704 2	0.676 5	0.726 1	0.693 4
ALbert-Tiny+DPCNN	0.786 7	0.753 5	<b>0.796 5</b>	<b>0.760 7</b>

3.5 模块与参数实验

上述实验中,模型 ALbert-Tiny + DPCNN 在对比实验中的准确率最高,基于该模型,本文进行了变体测试与参数实验。变体测试主要包括 2 个方向:对特征提取层输出的文本特征向量进行注意力加权、标签平滑归一化。接下来,可做阐释表述如下。

(1) ALbert-Tiny + DPCNN:对比实验中表现最优的模型。

(2) ALbert-Tiny + DPCNN + LSR:LSR 表示标签平滑归一化<sup>[15]</sup>,该技术通过平滑归一化方法扰动真实标签来解决过度拟合的问题。对标签的具体处理详见式(11):

$$y_k^{LS} = y_k(1 - \alpha) + \alpha/K \quad (11)$$

其中,  $y_k^{LS}$  为第  $k$  类标签平滑处理后的值;  $y_k$  为第  $k$  类标签的 one-hot 值;  $K$  为类别数;  $\alpha$  为超参数。

(3) ALbert-Tiny + DPCNN+ Att:注意力原理不依赖于任何框架,本文搭建了 2 层全连接层的 Att 注意力模块,通过该模块来捕获重要特征,减小无用特征,提升特征提取层的文本语义表征能力。具体如图 3 所示。

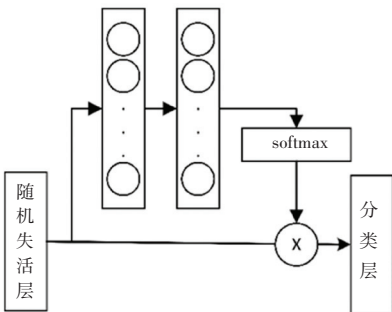


图 3 特征提取层使用的注意力模块

Fig. 3 Attention module

(4) ALbert-Tiny + DPCNN+ Att+ LSR:结合了(2)、(3)中的 Att 与 LSR 两种方式。

表 4 为变体模型的对比实验结果,从中可以得到以下结论:

(1)在特征提取层中使用注意力机制在一定程

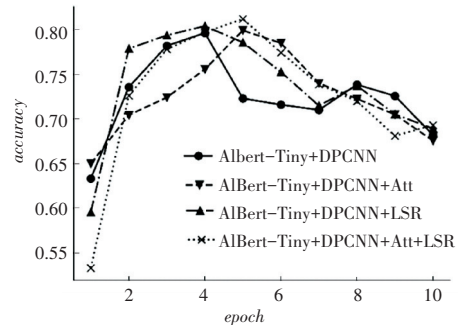
度上可以提高模型准确率,增强特征提取层的文本语义提取效果。

(2)标签平滑正则化在一定程度上增强了模型泛化能力,加快了模型的收敛。图 4(a)、图 4(b)分别为模型训练过程中准确率与加权  $F_1$  的变化曲线,明显可见使用 ALbert-Tiny 进行词嵌入时,在 epoch 等于 4 或 5 时能达到较好效果。

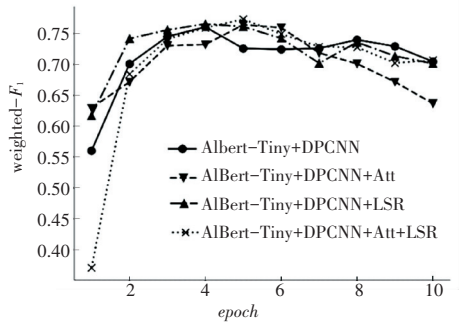
表 4 ALbert-Tiny+DPCNN 及其变体的准确率与加权  $F_1$  值

Tab. 4 Accuracy and weighted- $F_1$  value of ALbert-Tiny+DPCNN and its variants

模型	accuracy	weighted - $F_1$
ALbert-Tiny + DPCNN	0.796 5	0.760 7
ALbert-Tiny + DPCNN + Att	0.799 4	0.764 3
ALbert-Tiny + DPCNN + LSR	0.804 2	0.765 7
ALbert-Tiny + DPCNN + Att + LSR	<b>0.812 2</b>	<b>0.773 4</b>



(a) accuracy



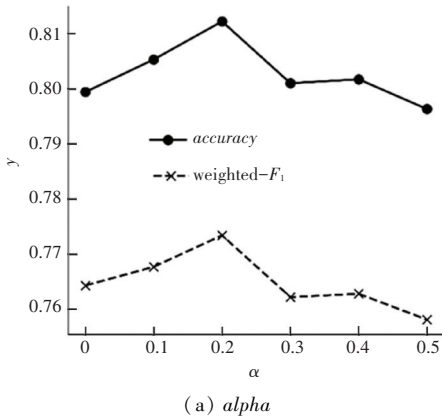
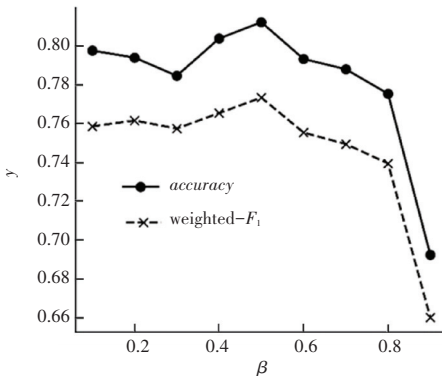
(b) weighted -  $F_1$

图 4 准确率与加权  $F_1$  值变化曲线

Fig. 4 Change curve of accuracy and weighted -  $F_1$  value

图 5(a)、图 5(b)分别记录了 ALbert-Tiny +

DPCNN + Att + LSR 的准确率  $Acc$  与加权  $F_1$  值 (Weighted- $F_1$ ) 随标签平滑归一化参数  $alpha$ 、随机失活层的失活系数  $beta$  的变化过程。标签平滑归一化参数  $alpha$  能够用来提高模型的泛化能力,从图 5(a)可见,当  $alpha$  在 0.2 附近时,模型的表现效果较好。随机失活系数  $beta$  是防止过拟合的优化参数,并从图 5(b)中可知,当  $alpha = 0.2$ 、 $beta$  值在 0.5 附近时,模型的表现效果较好。

(a)  $alpha$ (b)  $beta$ 图 5 准确率与加权  $F_1$  值随  $alpha$ 、 $beta$  变化的曲线Fig. 5 The curve of accuracy and weighted- $F_1$  values changing with  $alpha$  and  $beta$ 

## 4 结束语

本文提出了基于 Albert-Tiny-DPCNN 的分类模型,并将其应用于案件事实标签预测任务。该模型以案件事实为输入,以判决结果的倾向性类别为真实标签,通过对案件事实进行单标签多分类来实现类别预测。基于民间借贷一审判决书数据集的实验结果证明,本文提出的模型准确率能够达到 79.65%,高于其他对照组。此外,本文使用注意力机制加强了重要特征信息,减小了噪声与无用特征的影响,并用标签平滑归一化提升模型泛化能力,模型的准确率达到 81.22%,提升了 1.97%。

虽然,本文提出的模型在实验数据集上拥有较好的表现,但若想在实际应用场景中使用依旧存在以下改进之处。第一,深度学习中的网络模型需要大量数据的支撑,本文数据集的样本数量亟待扩充,模型泛化能力有待加强;第二,目前,NLP 领域中除 Albert 外还有 ELMO、MPNet 等预训练语言模型,因此下一步工作可以尝试深入探讨其他模型,另外采用多特征融合或结合传统机器学习的方法可能取得不错的效果。

## 参考文献

- [1] 吴叶乾, 韩青松. 对“同案同判”的全面理解及其实现[J]. 浙江理工大学学报(社会科学版), 2020, 44(01): 75-80.
- [2] 王文广, 陈运文, 蔡华, 等. 基于混合深度神经网络模型的司法文书智能化处理[J]. 清华大学学报(自然科学版), 2019, 59(07): 505-511.
- [3] 王业沛, 宋梦姣, 王譔, 赵志宏. 基于深度学习的判决结果倾向性分析[J]. 计算机应用研究, 2019, 36(02): 335-338.
- [4] 王宁, 李世林, 刘堂亮, 赵伟. 基于注意力机制的 BiGRU 判决结果倾向性分析[J]. 计算机系统应用, 2019, 28(03): 191-195.
- [5] CAMACHO-COLLADOS J, PILEHVAR M T. From word to sense embeddings: A survey on vector representations of meaning [J]. Journal of Artificial Intelligence Research, 2018, 63: 743-788.
- [6] LI Qian, PENG Hao, LI Jianxin, et al. A survey on text classification: From shallow to deep learning [J]. arXiv preprint arXiv:2008.00364, 2020.
- [7] 王立梅, 朱旭光, 汪德嘉, 等. 基于深度学习的民事案件判决结果分类方法研究[J]. 计算机科学, 2021, 48(08): 80-85.
- [8] 孟令慈. 基于 Bert-LSTM 模型的裁判文书分类的研究[D]. 华东交通大学, 2021.
- [9] MIKOLOV T, CHEN Kai, CORRADO G, et al. Efficient estimation of word representations in vector space [C] // International Conference of Learning Representation. Scottsdale, Arizona; IEEE, 2013: 1-12.
- [10] LAN Zhenzhong, CHEN Mingda, GOODMAN S, et al. Albert: A lite bert for self-supervised learning of language representations [J]. arXiv preprint arXiv:1909.11942, 2019.
- [11] KIM Y. Convolutional Neural Networks for sentence classification [J]. arXiv preprint arXiv: 1408.5882, 2014.
- [12] LAI Siwei, XU Liheng, LIU Kang, et al. Recurrent convolutional neural networks for text classification [C] // Twenty-ninth AAAI Conference on Artificial Intelligence. Austin, Texas, USA; AAAI, 2015: 2267-2273.
- [13] JOHNSON R, ZHANG Tong. Deep pyramid convolutional neural networks for text categorization [C] // Proceedings of the 55<sup>th</sup> Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Vancouver: ACL, 2017: 562-570.
- [14] LIN T Y, GOYAL P, GIRSHICK R, et al. Focal loss for dense object detection [C] // Proceedings of the IEEE International Conference on Computer Vision. Venice; IEEE, 2017: 2980-2988.
- [15] MÜLLER R, KORNBLITH S, HINTON G E. When does label smoothing help? [J]. arXiv preprint arXiv:1906.02629, 2019.