基于 CNN 的安防数据相似重复记录检测模型

王 巍[1,2,3] 刘 阳[1,2] 洪惠君[1,2] 梁雅静[1,2]

[1](河北工程大学信息与电气工程学院 河北 邯郸 056038)

[2](河北省安防信息感知与处理重点实验室 河北 邯郸 056038)

[3](江南大学物联网工程学院 江苏 无锡 214122)

摘 要 安防行业的结构化数据中存在大量的相似重复记录,传统的相似重复记录检测算法的识别率很难满足安防行业的实际需求。针对这种情况,引入了卷积神经网络模型,设计两种以 LeNet-5 模型为基础的改进模型,一种是输入为词向量矩阵的模型,另一种是输入为相似度矩阵的模型。实验表明,输入为词向量矩阵的模型的精确率和召回率均达到了 96% 以上,输入为相似度矩阵的模型的精确率和召回率高达 98%,并且 K 折交叉验证的结果说明模型具有较强的泛化能力。

关键词 安防行业 数据清洗 相似重复记录检测 CNN LeNet-5

中图分类号 TP311 **文献标志码** A DOI:10.3969/j.issn.1000-386x.2023.02.004

APPROXIMATELY DUPLICATE RECORD DETECTION MODEL FOR SECURITY DATA BASED ON CNN

Wang Wei[1,2,3] Liu Yang[1,2] Hong Huijun[1,2] Liang Yajing[1,2]

[1](*School of Information & Electrical Engineering, Hebei University of Engineering, Handan 056038, Hebei, China*)

[2](*Hebei Key Laboratory of Security & Protection Information Sensing and Processing, Handan 056038, Hebei, China*)

[3](*School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, Jiangsu, China*)

Abstract There are a lot of approximately duplicate record in the structured data of security industry. The recognition rate of traditional approximately duplicate record detection algorithm is difficult to meet the actual demand of security industry. In order to solve the above problems, a convolutional neural network model was introduced and two improved models based on LeNet-5 model were designed. One was the model with input as word embedding matrix, the other is the model with input as similarity matrix. The experiments show that the precision rate and recall rate of the model with input as word embedding matrix reach more than 96%. And the precision rate and recall rate of the model with input as a similarity matrix reach up to 98%. The experimental results of K-fold cross validation show that both models have strong generalization ability.

Keywords Security industry Data cleaning Approximately duplicate record detection CNN LeNet-5

0 引 言

随着信息化时代的飞速发展,每个行业每天都会产生大量的数据,安防行业也不例外。在安防行业中,部分结构化的信息还需要以手工录入、人为修改的方式进行[1]。这种方式不但效率低,而且会产生许多的脏数据,会直接影响到各类事件的后续处理效率。这些脏数据的表现形式主要有相似重复记录、缺失数据、冲突数据和错误数据等几种类型[2]。其中数据源中存在相似重复记录是最严重的问题之一,严重时可能会导致数据失效。因此相似重复记录的检测及清除一直

收稿日期:2020 - 07 - 02。国家自然科学基金项目(61802107);教育部-中国移动科研基金项目(MCM20170204);江苏省博士后科研资助计划项目(1601085C)。**王巍**,副教授,主研领域:公共安全物联网,智能控制理论与应用。**刘阳**,硕士生。**洪惠君**,硕士生。**梁雅静**,硕士生。

都是研究的热点。

当前常见的相似重复记录检测算法主要有编辑距离算法(Levenshtein Distance)、Smith-Waterman 算法、N-gram 算法和机器学习算法等。陈俊月等[3]提出了基于词向量的编辑距离算法,改进了传统的编辑距离算法和 Jaccard 系数,将编辑距离的删除和插入两种编辑操作修改为空字符与目标字符的替换操作,通过词向量计算每种编辑操作的相似度并累加,得出改进的编辑距离相似度。文献[4]改进了编辑距离算法,使用同义词词典替换了传统的语法变异的方法,对于存在大量语法错误和拼写错误的问题的数据集,可有效提高算法的精度。文献[5]利用依赖图的传递闭包和循环倾斜提出了一种 Smith-Waterman 算法并行化的方法,将精确传递闭包的方法替代为近似传递闭包,在现代多核计算机中加速效果显著。文献[6]对比了三种基于统一计算设备架构(Compute Unified Device Architecture,CUDA)的 Smith-Waterman 算法的并行计算方法,分别是粗粒度加速方法、细粒度加速方法和重新定义递归公式的方法。其中:粗粒度加速方法可利用 GPU 同时处理多个对齐,但是不同线程处理长度不同的序列的时间不同,导致处理短序列的线程等待时间较长;细粒度加速方法可以利用多线程计算 DP 矩阵,减少了中间计算的时间,但是在计算开始和结束的阶段都需要花费大量的时间;重新定义递归公式的方法重新定义了 Smith-Waterman 算法的递归公式,将算法时间复杂度从 $O(mn)$ 降低到了 $O(m\log 2n)$,其中 m 和 n 表示任意两条记录的长度,提高了算法的效率,但不能对差距的扩大给出不同的罚分。王常武等[7]提出了一种加权的 N-Gram 相似重复元数据记录检测算法,对于不同的标签在相似重复记录检测的过程中的重要程度赋予不同的权重,依据权重使用加权的 N-gram 算法进行相似度计算,实验表明,加权后的 N-gram 算法优于普通的 N-gram 算法。陈亮等[8]使用分块技术根据字段对数据排序并分块,使用滑动窗口的思想提升检测效率,然后将各字段得出的分块一起聚类,对重复率较大的分块对首先比对,并放弃聚类效果较差的分块,最终通过降低聚类次数和放弃聚类效果较差的分块的方式提升了效率和准确率。

近年来机器学习成了研究的热点,几乎在各行业都有应用案例,也有研究人员运用机器学习算法来检测相似重复记录。相似重复记录的检测在机器学习中可以看作分类问题中的二分类问题。吕国俊等[9]提出了一种基于多目标蚁群优化的支持向量机的相似重复记录检测方法,该方法将相似重复记录描述为一个二分类问题,同时考虑到相似重复记录中重复样本很少的问题,只用不相似重复记录样本进行训练。文中的实验结果验证单类支持向量机算法的有效性。张攀[10]将 BP 神经网络运用到相似重复记录的检测当中,用两条记录相同字段间的编辑距离组成的向量作为 BP 网络的输入数据,并为其添加上标签训练出 BP 网络,然后使用训练出的 BP 神经网络检测两条记录是否为相似重复记录,实验结果表明,在数据生成器“febrl”生成的数据中,基于 BP 神经网络的相似重复记录检测方法的准确率可达 97% 以上。孟祥逢等[11]首先融合 Jaro 算法与 TF-IDF 算法计算字段相似度,然后使用经过遗传算法优化的神经网络计算两条记录的相似度,可以有效提高检测的准确率和精度,但在中文数据中的效果还有待商榷。

尽管上述相似重复记录检测算法的精确率(Precision)和召回率(Recall)等指标都已经达到了较高的水平,但是上述数据大部分是开源的数据集,实际应用中的数据集存在更复杂的情况。在安防数据库中,由于不同信息录入员的不同习惯,存在大量空值、错误值、串字段错误和各种符号等干扰信息,对相似重复记录的检测产生了一定的影响,同时安防数据中部分相似重复记录肉眼辨别都需要较长时间。例如表 1 中前两条记录表述不同,但实际含义相同,是相似重复记录,而后两条记录虽然仅存在个别字不同,但其是两个不同的加油站,不是相似重复记录。因此,安防数据相似重复记录的检测难度比网络上开源的数据集要高,需要开发一种适用于安防数据的相似重复记录检测模型,检测出尽可能多的相似重复记录对,同时还需要较强的泛化能力。卷积神经网络(Convolutional Neural Networks,CNN)是深度学习的代表算法之一,在深度学习中的研究最为广泛[12]。近年来,CNN 开始应用于文本分类等自然语言处理领域,并且取得了一定的成果。例如,文献[13 - 14]将 CNN 应用于文本情感分类;宋岩等[15]使用 CNN 对文本进行分类;肖琳等[16]使用 CNN 对文本进行多标签分类;张璞等[17]结合 CNN 与微博文本特征对用户性别进行分类等。

表 1 安防数据相似重复记录示例

名称	类型	地址
兴业银行:霸州支行	金融	胜芳芳清道 26 号
霸州支行	兴业银行	胜芳镇芳清道 26 号
中石油风宁县第八十三加油站	石油	丰宁县小坝子乡曹碾沟村水泉停车区南
中石油风宁县第八十四加油站	石油	丰宁县小坝子乡曹碾沟村水泉停车区北

因此,本文设计一种基于 CNN 的安防数据相似重

复记录检测模型。通过对在手写数字识别上拥有不错效果的 LeNet-5 进行改进,得出了两种适用于安防数据的相似重复记录模型,一种是以词向量矩阵为输入的 CNN 模型(Word Embedding Matrix CNN,WE-CNN),另一种是以相似度矩阵为输入的 CNN 模型(Similarity Matrix CNN,SIM-CNN)。然后分别对这两种模型进行了介绍,并通过了 K 折验证,使用精确率、召回率和 F1 值(F1 Score)等评价指标对模型进行了评价。

1　数据预处理

为了提高模型的效率和可靠性,在训练模型之前,需要对数据进行预处理,并且需要将每条记录中的每个词都转换为模型可以识别的词向量(Word Embedding)。根据安防数据存在的问题以及安防数据的特点,本文将数据预处理主要步骤分为分词和词向量生成两部分,具体流程如下:

(1) 字段选择。手动选择数据库中对于任意两条记录是否为相似重复记录影响较大的字段,并且删除数据库中的空记录等无效的记录。

(2) 分词。对记录中的每一个字段进行中文分词,并删除停用词、标点、特殊字符等内容。本文采用"jieba"分词工具进行分词。将每条记录分词后的内容按照字段顺序组成一个列表,也就是一个列表对应一条记录,则第 i 个列表的长度 $L(i)$ 表示为:

$$L(i) = \sum_{f=1}^{n} l_{if} \tag{1}$$

式中:n 为字段数;l_{if} 为第 i 个列表的第 f 个字段分词后的词语的个数。

(3) 训练词向量。对分词后的所有列表进行词向量的训练,得出每一个词的词向量 c。本文采用 Google 的"Word2vec"工具训练词向量,主要参数如表 2 所示。其语料库为本文的实验数据某安防报警网络公司用户数据。

表 2　"Word2vec"词向量训练工具主要参数

参数名称	参数选择
训练算法	Skip-gram 算法
词向量维度	100
窗口大小	5
词频过滤	2
负采样	使用负采样

表 2 中的 Skip-gram 算法使用中心词预测周围词语,根据预测结果修正中心词词向量。通过对安防领域数据的验证,大部分字段为中文短句或者长词组,每个词与前后词的关联较大,并且这些短句或者长词组对两条记录是否为相似重复记录影响较大,因此对于安防领域数据,可使用"Word2vec"的 Skip-gram 算法训练词向量。

(4) 合成矩阵。将每一个记录中的所有词语对应的词向量组成一个向量组,该向量组就是词向量矩阵。假设第 i 条记录的词向量矩阵是 $\boldsymbol{V}_i$,则该矩阵表示为:

$$\boldsymbol{V}_i = [\boldsymbol{c}_1, \boldsymbol{c}_2, \cdots, \boldsymbol{c}_{L(i)}] \tag{2}$$

使用 Word2vec 中的相似度计算算法,计算出两条记录 m 和 n 中所有词之间的相似度,组成相似度矩阵 $\boldsymbol{S}$,表示为:

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{s}_{11} & \boldsymbol{s}_{12} & \cdots & \boldsymbol{s}_{1L(n)} \\ \boldsymbol{s}_{21} & \boldsymbol{s}_{22} & \cdots & \boldsymbol{s}_{2L(n)} \\ \vdots & \vdots & & \vdots \\ \boldsymbol{s}_{L(m)1} & \boldsymbol{s}_{L(m)2} & \cdots & \boldsymbol{s}_{L(m)L(n)} \end{bmatrix} \tag{3}$$

(5) 统一矩阵大小。统一各词向量矩阵的矩阵大小和各相似度矩阵的矩阵大小,根据分词后的每一个列表的长度 $L(i)$($i \in \mathbf{N}$ 且 $i \in [0,n)$,n 是数据库中删除无效记录后的记录数),选择合适的矩阵大小,不足的部分补充 0,超出的部分删除。如果选择的矩阵大小太小,无法包含记录的有效信息,相反,过大会降低计算的效率。图 1 为本文选用的安防数据记录对应的记录长度比例分布图,横坐标表示记录长度,纵坐标表示小于等于某长度的记录与全部记录的数量比。可以看出 95% 左右的记录长度都在 60 以下,因此本文的词向量矩阵大小为 60 × 100,相似度矩阵大小为 60 × 60。

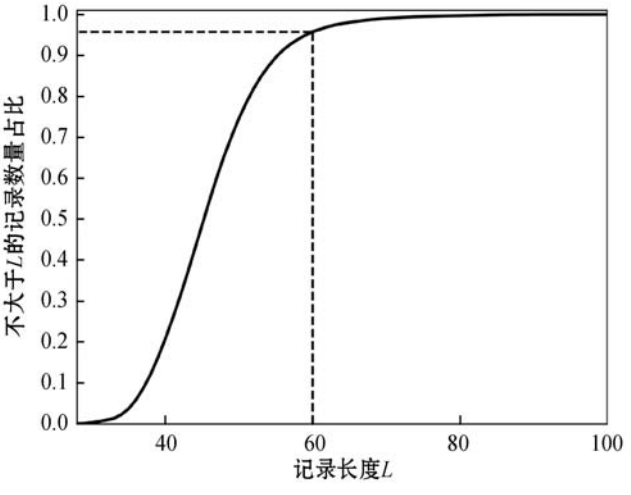

图 1　安防数据记录长度比例分布

(6) 标记数据。标记两条记录 m 和 n 是否为相似重复记录,并将 m 和 n 词向量矩阵与标签组成带标签的词向量矩阵数据。将两条记录 m 和 n 组成的相似度矩阵与标签组成带标签的相似度矩阵数据。

上述流程如图 2 所示。

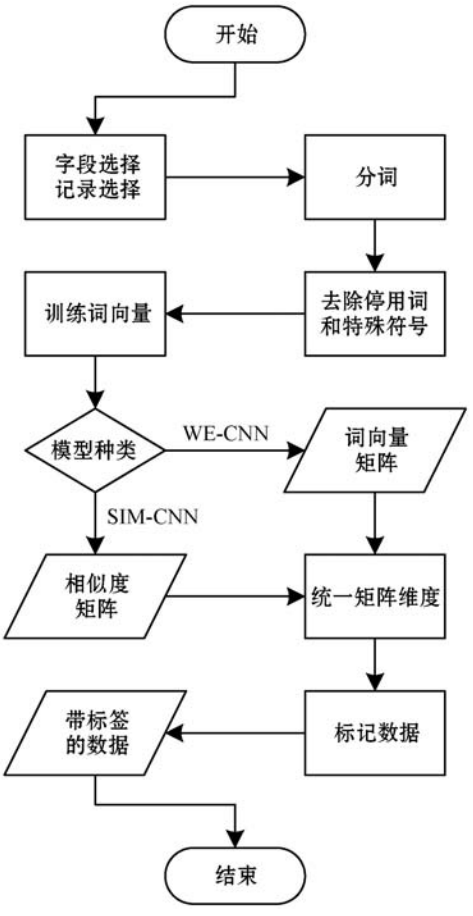

图 2 数据预处理流程

2 模型设计

2.1 LeNet-5 模型

Fukushima[18] 最早提出了卷积神经网络,除去输入层和输出层,卷积神经网络通常还包含若干个卷积层、池化层和全连接层。LeNet-5[19] 是出现时间最早的卷积神经网络之一,由 Yann LeCun 提出,最早应用于手写体识别。LeNet-5 的结构如图 3 所示。

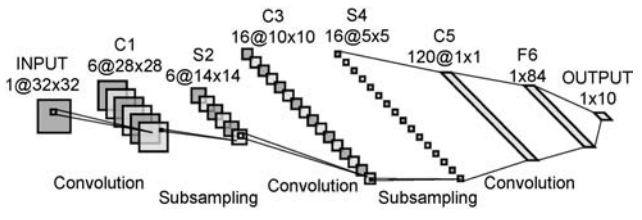

图 3 LeNet-5 结构

LeNet-5 一共有 8 层,包括:3 个卷积层 C1、C3 和 C5;两个池化层 S2 和 S4;一个全连接层 F6;一个输入层 INPUT 和一个输出层 OUTPUT。其中 INPUT 层的输入是 1 幅维度为 32 的图片;C1 层通过 6 个 5 ×5 大小的卷积核得出 6 幅维度为 28 的特征图;S2 层将上一层特征图中 2 ×2 范围内的数值相加并乘一个系数加一个偏置,得出 6 幅维度为 14 的特征图,这个过程称为二次采样(Subsampling),其中的系数和偏置可训练;C3 层采用部分映射的方式将 S2 层的 6 幅特征图通过 5 ×5 大小的卷积核转换为 16 幅 10 ×10 的特征图,表 3 展示了具体对应的方案,其中的数字为特征图(通道)号;S4 层通过二次采样,得出 16 幅 5 ×5 的特征图;C5 层用 5 ×5 的卷积核对 5 ×5 的特征图进行卷积操作,将 S4 层的 16 幅特征图转换为 120 幅 1 ×1 的特征图;F6 层包括 84 个单元,并且完全连接到 C5 层。OUTPUT 层有 10 个欧氏径向基函数单元,表示十个手写数字,可得到图片是哪个手写数字。

表 3 S2 与 C3 的连接方案

S2 层	C3 层
0	0、4、5、6、9、10、11、12、14、15
1	0、1、5、6、7、10、11、12、13、15
2	0、1、2、6、7、8、11、13、14、15
3	1、2、3、7、8、9、12、14、15
4	2、3、4、8、9、10、12、13、15
5	3、4、5、9、10、11、13、14、15

2.2 改进的 LeNet-5 模型

(1) WE-CNN 模型。与复杂的图像识别相比,文本识别的复杂度要低一些。而 LeNet-5 可以识别较为简单的手写体图片,所以将 LeNet-5 应用于相似重复记录的识别具有一定的可行性。为了让 LeNet-5 模型适用于文本识别,并且可以有效解决安防数据相似重复记录检测存在的无法填补的空值、串字段等难点,本文设计以记录的词向量矩阵为输入的 CNN 模型 WE-CNN,模型在 LeNet-5 的基础上进行改进,改进后的模型如图 4 所示,具体做了以下几个部分的改进。

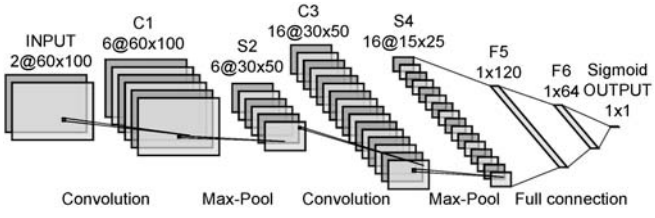

图 4 WE-CNN 模型结构

在 LeNet-5 中,输入维度为 32 ×32,为了保证得到更全的细节特征,提高模型的识别率,将输入的维度修改为与数据预处理中描述的 60 ×100,同时因为要判断两条记录的相似度,所以输入的通道数为 2。

为了简化算法,将 S2 层与 C3 层之间的连接方案改为全映射;把 C5 层改为全连接层并改名为 F5,这样可以更好地将特征矩阵中的信息传递下去,这里涉及的特征矩阵对应 LeNet-5 中的特征图;C1 层和 C3 层均采用 3 ×3 的卷积核,并且将卷积操作的填补

(padding)设置为 1,在进行卷积操作时可以不改变特征矩阵的维度,在简化各层之间的维度计算的同时可以保留更全的信息;将 S2 层和 S4 层中的二次采样修改为最大池化(Max-Pool),可以保留记录中更多的边缘特征。

由于相似重复记录检测问题属于二分类问题,故将模型的输出从十个改为一个,并使用 Sigmoid 函数,将输出值转化为一个介于 0 到 1 之间的概率值,概率值越接近 1,表示输入的两条记录为相似重复记录的可能性越大,反之不是相似重复记录的可能性越大,区分是否是相似重复记录的阈值通常取 0.5。

为了防止出现过拟合的现象,在 S2 层与 C3 层之间、S4 层和 F5 层之间、F5 层和 F6 层之间、F6 层和 OUTPUT 层之间均加入 Dropout 方法。Dropout 方法可以随机丢弃 $p\%$ 的神经元,使用剩下的神经元进行训练,可以防止过拟合和增强模型的泛化能力。为了减轻梯度消失问题对模型的影响,在 S2 层、S4 层、F5 层、F6 层和 F7 层均设置一个 tanh 函数。

使用 Adam 算法作为模型的优化算法。Adam 算法结合了 Momentum 算法和 RMSprop 算法中的更新方向方法和计算衰减系数方法,式(4)为 Adam 算法的算法策略。

$$\begin{cases} m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \\ v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \\ \hat{m}_t = \dfrac{m_t}{1-\beta_1^t} \\ \hat{v}_t = \dfrac{v_t}{1-\beta_2^t} \\ W_{t+1} = W_t - \eta \dfrac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon} \end{cases} \tag{4}$$

式中:m_t 和 v_t 分别是第 t 次迭代的一阶矩估计和二阶矩估计;β_1 和 β_2 为动力值,通常 $\beta_1 = 0.9$,$\beta_2 = 0.999$;g_t 表示第 t 次迭代时的梯度值;$\hat{m}_t$ 和 $\hat{v}_t$ 是 m_t 和 v_t 的修正值,使用修正值可以消除初始化的偏差;W_t 是第 t 次迭代的模型参数;η 为学习速率;ε 是一个很小的数,通常为 10^{-8},目的是防止分母为 0。

在二分类问题中,可以采用适用于二分类问题的交叉熵损失函数函数(Cross Entropy Error Function)作为模型的损失函数,式(5)为该函数的表达式。

$$\begin{cases} l(x,y) = \{l_1, l_2, \cdots, l_i, \cdots, l_n\} \\ l_i = -[y_i \cdot \ln(x_i) + (1-y_i) \cdot \ln(1-x_i)] \end{cases} \tag{5}$$

式中:n 是批量大小(batch size);x 是模型实际输出的值;y 是标签上的值,也就是目标值。

(2) SIM-CNN 模型。在安防行业中,某些数据会有极高的质量要求,为了进一步提高数据的质量,在 WE-CNN 模型的基础上,提出以相似度矩阵为输入的 CNN 模型 SIM-CNN。从理论上来说,与 WE-CNN 模型相比,在同样的数据源下,SIM-CNN 模型应该具有更高的识别率,但是 SIM-CNN 模型所需的相似度矩阵生成的时间较长。

除了输入层,SIM-CNN 与 WE-CNN 模型的其余部分完全相同。SIM-CNN 的输入为通道数为 1 的相似度矩阵,根据数据预处理中的描述,将输入的相似度矩阵维度设置为 60,模型的结构如图 5 所示。

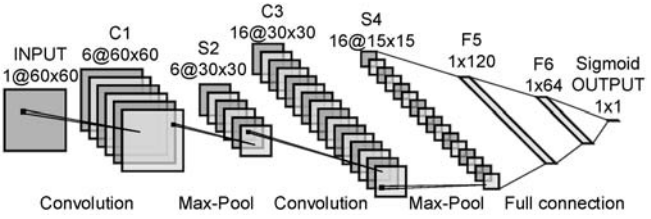

图 5　SIM-CNN 模型结构

3　实验验证

本次实验首先对实验数据、实验环境和实验中用到的一些评价方法进行了介绍;然后对模型中的部分参数进行了选择分析,并对模型的效果、泛化能力进行了验证;最后与 BP 神经网络进行了对比。

3.1　实验数据说明

实验采用某安防网络报警公司的用户数据作为实验数据,一共有 32 480 条记录,120 个字段。经过筛选,得到了 30 513 条有意义的记录,26 个对相似重复记录的识别影响较大的字段。经过初步判断,实验数据中有 6 100 对左右的相似重复记录。

本次实验选取了 11 000 对记录作为训练数据和测试数据,其中相似重复记录和不是相似重复记录的数据各 5 500 对。

3.2　实验环境

实验的硬件环境配置如表 4 所示。

表 4　实验硬件环境

硬件名称	硬件参数
处理器	英特尔 Core i5-6300HQ,四核四线程
RAM	海力士 DDR4 2 133 MHz,8 GB
硬盘	三星 MZVLV128HCGR-000L2,128 GB 固态硬盘
显卡	Nvidia GeForce GTX 950M,4 GB

本实验采用了 CUDA 并行计算架构,实验的软件环境如表 5 所示。

表 5 实验软件环境

软件名称	软件参数
操作系统	Windows 10 64 位,家庭中文版
深度学习框架	PyTorch 1.1.0
CUDA 版本	9.0
编程语言	Python 3.7.3
词向量生成工具	genism Word2vec
分词框架	jieba 0.42.1
数据库管理系统	SQL Server 13.00.1601

3.3 实验评价方式

为了评价模型的优劣,引入了精确率、召回率和 F1 值等评价指标。精确率、召回率和 F1 值分别可以用式(6)-式(8)表示。

$$pre = \frac{TP}{TP + FP} \tag{6}$$

$$rec = \frac{TP}{TP + FN} \tag{7}$$

$$F1 = \frac{2 \cdot pre \cdot rec}{pre + rec} \tag{8}$$

假设将不是相似重复记录记为正类,是相似重复记录记为负类,则式(6)-式(8)中:*TP* 表示实际上和预测的都不是相似重复记录的数据对的个数,记为真正类;*FN* 表示实际上不是相似重复记录,预测是相似重复记录的数据对的个数,记为假负类;*FP* 表示实际上是相似重复记录,预测不是相似重复记录的数据对的个数,记为假正类;*TN* 表示实际上和预测的都是相似重复记录的数据对的个数,记为真负类。

为了验证模型的可靠性,引入 K 折交叉验证的方法对模型进行分析。K 折交叉验证是将数据分成 *K* 组,使用任意一组作为测试集,剩余的 *K*-1 组作为训练集,迭代 *K* 次,让每一组数据都做过一次测试集,每次迭代的测试集中的数据不会出现在当次迭代的训练数据中。

3.4 实验结果和分析

(1) 学习速率选择实验。在批量大小 *batch-size* 都为 64,迭代次数 *epoch* 都为 60,以及其他参数都相同的情况下,从 0.01、0.005、0.001、0.000 5 和 0.000 1 五个学习速率当中确定合适的学习速率。从理论上来说,在各参数都相同的条件下,如果学习速率较大,可能会出现发散的情况,如果学习速率越小,收敛速度越慢。本实验中,在学习速率在 0.01 时,两个模型出现了损失发散的情况,在学习速率为 0.005 时,WE-CNN 模型存在低概率的发散,故舍弃这些导致损失发散的学习速率。两个模型在不同学习速率下的各项指标如图 6 和图 7 所示。

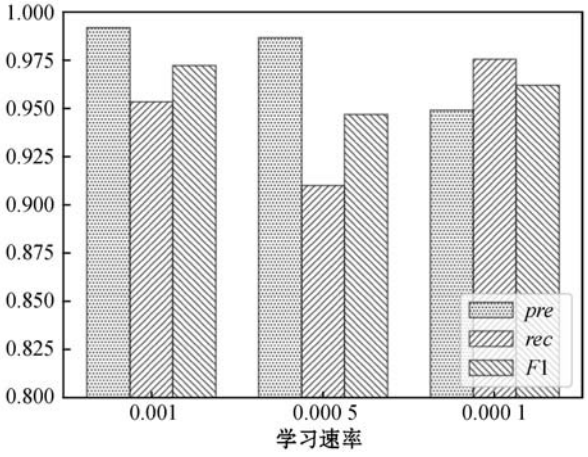

图 6 WE-CNN 模型不同学习速率下的表现

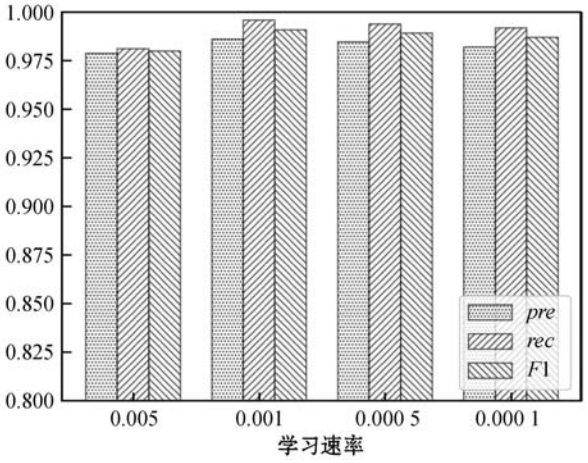

图 7 SIM-CNN 模型不同学习速率下的表现

从图 6 中可以看出,WE-CNN 模型的学习速率为 0.001 时的整体效果最好,其 F1 值最大,但是召回率较低;从图 7 中可以看出,当 SIM-CNN 模型在学习速率为 0.001、0.000 5 和 0.000 1 时,精确率、召回率和 F1 值都很接近,并且都在 0.975 以上,不过学习速率等于 0.001 时的 SIM-CNN 模型略优于其他学习速率的模型。这是因为随着迭代次数的增加,学习速率为 0.001 的模型可以最先达到最优结果。所以在本实验的测试数据集中,WE-CNN 模型和 SIM-CNN 模型的学习速率在 0.001 时模型的表现最优。

(2) Dropout 对模型的影响。表 6 介绍了是否使用 Dropout 优化方法对模型的影响。可以得出 WE-CNN 模型在不使用 Dropout 优化方法时会对模型的识别能力产生较大的影响,因为每次训练都是使用相同的模型,所以出现了过拟合现象。而使用 Dropout 可以使神经元随机失活,每次训练的模型都不同,因此不容易出现过拟合现象,识别率相对更高。SIM-CNN 模型在不使用 Dropout 优化方法时同样会对模型的识别能力产生影响,所以使用 Dropout 优化方法可以有效防止过拟

合,增强模型的识别能力。

表 6　Dropout 优化方法对模型的影响

模型	Dropout	*TP*	*FN*	*FP*	*TN*	*pre*	*rec*	*F*1
WE-CNN	有	480	20	4	496	0.976	0.960	0.976
	无	390	110	50	450	0.840	0.780	0.830
SIM-CNN	有	498	2	4	496	0.994	0.996	0.994
	无	485	15	3	497	0.982	0.970	0.982

(3) K 折交叉验证。根据 K 折交叉验证的规则,将本次实验的 11 000 个样本,分成了 11 组数据,每组数据中都有 500 个正类样本和 500 个负类样本。从第 1 组样本开始,依次将该组数据作为测试集,并将剩余的数据作为第 1 组的训练集。将学习速率设置为 0.001,批量值 *batch-size* =64,迭代次数 *epoch* 设置为 60,阈值设置为 0.5,通过 WE-CNN 模型在所有的训练集进行训练,并使用每个训练集对应的测试集对模型进行测试,使用式(5)所示的交叉熵损失函数计算损失,得出 K 折验证中每个测试集的精确率、召回率和 F1 值。WE-CNN 模型的 K 折交叉验证的各项指标如表 7 所示。

表 7　WE-CNN 模型 K 折交叉验证各项指标

组号	*TP*	*FN*	*FP*	*TN*	*pre*	*rec*	*F*1
1	496	4	2	498	0.996	0.992	0.994
2	497	3	1	499	0.998	0.994	0.996
3	500	0	5	495	0.990	1.000	0.995
4	489	11	46	454	0.914	0.978	0.945
5	492	8	29	471	0.944	0.984	0.964
6	439	61	14	486	0.969	0.878	0.921
7	497	3	5	495	0.990	0.994	0.992
8	451	49	0	500	1.000	0.902	0.948
9	491	9	0	500	1.000	0.982	0.991
10	478	22	2	498	0.996	0.956	0.976
11	486	14	2	498	0.996	0.972	0.984
均值	483.3	16.7	9.6	490.4	0.981	0.967	0.973
偏差	19.1	19.1	14.1	14.1	0.027	0.038	0.024

可以看出,虽然第 6 组数据对应的模型的召回率只有 0.878,精确率和 F1 值也只有 0.969 和 0.921,出现这种情况可能是因为该组数据的测试集中存在个别的标记错误。但是从整体上看,大部分组的精确率、召回率和 F1 值都在 0.95 到 0.99 左右,处于一个范围较小的区间内,平均指标中最低的召回率也高达 0.967,偏差也仅有 0.038,F1 值为 0.973,偏差仅为 0.024。但是在实际情况中,相似重复记录的占比较少,而召回率可以表示数据中没有相似重复记录的极端情况,所以 WE-CNN 模型在极端情况下的平均识别率也能高达 0.967。因此,WE-CNN 模型具有较高的识别率和较强的泛化能力。

基于 WE-CNN 模型改进的 SIM-CNN 模型的 K 折交叉验证的各项指标如表 8 所示。

表 8　SIM-CNN 模型 K 折交叉验证各项指标

组号	*TP*	*FN*	*FP*	*TN*	*pre*	*rec*	*F*1
1	497	3	0	500	1.000	0.994	0.997
2	494	6	2	498	0.996	0.988	0.992
3	499	1	3	497	0.994	0.998	0.996
4	496	4	28	472	0.947	0.992	0.969
5	494	6	16	484	0.969	0.988	0.978
6	462	38	18	482	0.963	0.924	0.943
7	498	2	1	499	0.998	0.996	0.997
8	497	3	0	500	1.000	0.994	0.997
9	498	2	0	500	1.000	0.996	0.998
10	473	27	2	498	0.996	0.946	0.970
11	498	2	1	499	0.998	0.996	0.997
均值	491.5	8.5	6.5	493.5	0.987	0.983	0.985
偏差	11.6	11.6	9.2	9.2	0.018	0.023	0.017

其中 SIM-CNN 模型的 K 折交叉验证实验的每一组的训练数据和测试数据都与 WE-CNN 模型 K 折交叉验证实验中对应的组中的数据相同。各类超参数也与 WE-CNN 模型的 K 折交叉验证实验相同。从表 9 中可以看出,SIM-CNN 模型的各项指标要明显优于 WE-CNN,整体上都有了一定的提升。同样是最低的第 6 组数据对应的模型的召回率已经达到了 0.924,比 SIM-CNN 模型提升了 0.046。从整体上来看,大部分的精确率、召回率、F1 值都在 0.97 到 1 之间,处于一个很小的区间内,从平均上来看,精确率、召回率和 F1 值都达到了 0.98 以上,而偏差最大的召回率也仅有 0.023,F1 值也高达 0.985,偏差仅有 0.017。因此,与 WE-CNN 模型相比,SIM-CNN 模型具有更高的识别率和更强的泛化能力。

(4) 运行时间。数据量为本次实验所用的 11 000 对记录,*batch-size* 均设置为 64,迭代次数 *epoch* 均设置为 60,WE-CNN 模型和 SIM-CNN 模型的训练数据和测试数据都相同。两模型的整体运行时间如表 9 所示。

表 9 模型运行时间 单位:s

模型	数据预处理和加载时间	模型训练和测试时间	总时间
WE-CNN	81	992	1 073
SIM-CNN	556	717	1 273

从模型训练和测试时间上来看,SIM-CNN 模型的训练速度要比 WE-CNN 模型快,但是由于 SIM-CNN 模型的输入是两条记录对应的相似度矩阵,生成实验所需的所有相似度矩阵需要 556 s,而生成 WE-CNN 模型所需的所有词向量矩阵仅需 81 s。因此从整体上看,在本次实验的数据集中,WE-CNN 模型的整体运行时间为 1 073 s,比 SIM-CNN 模型的整体运行时间少 200 s。

(5) 模型对比。图 8 为相同的实验数据在 WE-CNN、SIM-CNN 和文献[10]所述的 BP 神经网络 3 种模型下的各项指标柱状图。

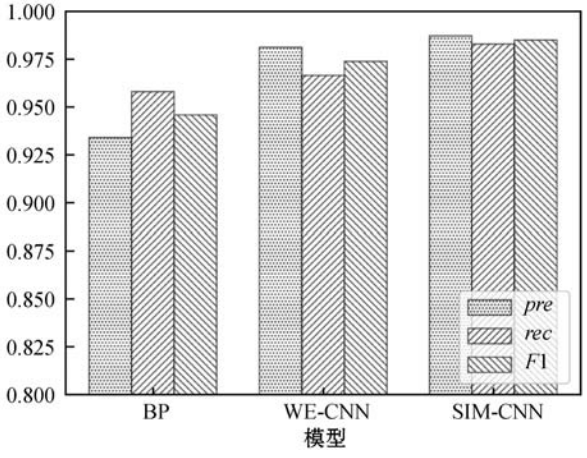

图 8 3 种模型指标对比

可以看出,在整体识别效果上 SIM-CNN 模型最好,其次是 WE-CNN 模型,最后是 BP 神经网络。这是因为 SIM-CNN 的输入为相似度矩阵,与 WE-CNN 的输入词向量矩阵相比,在都保留了原有信息的基础上输入更简洁,需要训练的参数更少,因此 SIM-CNN 模型的识别效果最优。而 BP 网络的输入是相同字段间的编辑距离值组成的向量,而安防数据中的部分相似重复记录的记录对相同字段间的编辑距离值较大,部分不是相似重复记录的记录对相同字段之间的编辑距离值较小,识别效果自然较差。

3.5 实验评价

综上所述,SIM-CNN 模型在上述实验数据集中的识别率和在没有相似重复记录的极端情况下的识别率(实验数据集中的召回率)均优于 WE-CNN 模型,但是 SIM-CNN 模型的数据生成和加载时间较长,导致整个训练和测试的时间要长于 WE-CNN 模型。如果 WE-CNN 模型可以达到识别率的要求,或者检测时长受限,可以使用 WE-CNN 模型进行相似重复记录的检测。如果对识别率等指标要求较高,或者检测时长宽松,可以使用 SIM-CNN 模型进行相似重复记录的检测。

4 结 语

对于任意两条记录来说,相似重复记录检测只有是相似重复记录和不是相似重复记录两种情况,故本文将相似重复记录检测问题视为了二分类问题。同时考虑到安防行业数据的相似重复记录检测要比网络上的开源数据的检测难度高,并且需要较高的识别率和较强的泛化能力。因此引入了 CNN 检测相似重复记录,提出以词向量矩阵为输入的 WE-CNN 模型和以相似度矩阵为输入的 SIM-CNN 模型。

WE-CNN 是以 LeNet-5 为基础进行改进的,现将改进内容总结为以下三部分。

1) 输入和输出层:将输入层设置为两个 60 × 100 的词向量矩阵;输出层增加 Sigmoid 函数。

2) 输入和输出层以外各层:将 S2 层与 C3 层之间设置为全映射;C5 层设置为全连层并改名为 F5;C1 和 C3 层的采用 3 × 3 的卷积核;S2 和 S4 层均设置为最大池化;在各层之间加入 Dropout 方法。

3) 优化器和损失函数:使用 Adam 优化器对模型进行优化;使用交叉熵损失函数对模型的损失进行计算。

将改进后的模型(WE-CNN)应用于安防数据相似重复记录检测,通过 K 折交叉验证,得出了平均精确率、召回率和 F1 值均在 0.96 以上的结果。

SIM-CNN 改进了 WE-CNN 的输入层,将一组词向量矩阵浓缩为一个相似度矩阵。在安防数据相似重复记录检测中,SIM-CNN 模型 K 折交叉验证的平均精确率、召回率和 F1 值均在 0.98 以上。SIM-CNN 模型在损失了数据预处理和生成的时间的同时,精确率、召回率和 F1 值相对于 WE-CNN 模型分别提高了 0.612%、1.656% 和 1.131%,偏差相对降低了 33.333%、39.474% 和 29.167%。

以上验证了两种模型都具有较高的稳定性和泛化能力,可以适用于安防数据相似重复记录检测。但是当前模型和模型生成的过程还存在需要进一步完善或者改进的问题,比如安防数据标记方法较难、安防数据中的相似重复记录的数量可能小于训练所需的相似重复记录的数量和安防数据相似重复记录清除算法的设

计等问题还需要解决。

参 考 文 献

[1] 苗清. 如何运用 AI 与大数据打造优质智能安防报警[J]. 中国安防,2019(12):94 - 97.

[2] 郝爽,李国良,冯建华,等. 结构化数据清洗技术综述[J]. 清华大学学报(自然科学版),2018,58(12):1037 - 1050.

[3] 陈俊月,郝文宁,张紫萱,等. 基于改进句子相似度的释义识别研究[J]. 计算机工程,2020,46(9):76 - 82.

[4] Niewiarowski A. Short text similarity algorithm based on the edit distance and thesaurus[J]. Technical Transactions. Fundamental Sciences,2016,113:159 - 173.

[5] Palkowski M, Bielecki W. Parallel tiled codes implementing the smith-waterman alignment algorithm for two and three sequences[J]. Journal of Computational Biology: A Journal of Computational Molecular Cell Biology,2018,25(10):1106 - 1119.

[6] Jing C, Zhang L. The review of the acceleration of Smith-Waterman algorithm by using CUDA-enable GPU[C]//2015 3rd International Conference on Machinery, Materials and Information Technology Applications,2015:758 - 766.

[7] 王常武,韩菁华,张付志. 一种相似重复元数据记录检测方法[J]. 计算机工程,2009,35(21):85 - 87.

[8] 陈亮,杜璐,胡康. 基于分块和滑窗技术的相似重复记录检测算法研究[J]. 计算机应用与软件,2019,36(4):262 - 267.

[9] 吕国俊,曹建军,郑奇斌,等. 基于多目标蚁群优化的单类支持向量机相似重复记录检测[J]. 兵工学报,2020,41(2):324 - 331.

[10] 张攀. 面向重复记录检测的数据清洗算法的研究[D]. 西安:西安电子科技大学,2018.

[11] 孟祥逢,鲁汉榕,郭玲. 基于遗传神经网络的相似重复记录检测方法[J]. 计算机工程与设计,2010,31(7):1550 - 1553.

[12] Gu J, Wang Z, Kuen J, et al. Recent advances in convolutional neural networks[J]. Pattern Recognition, 2018,77:354 - 377.

[13] 李丽华,胡小龙. 基于深度学习的文本情感分析[J]. 湖北大学学报(自然科学版),2020,42(2):142 - 149.

[14] 李卫疆,伊靖. 基于扩展特征矩阵和双层卷积神经网络的微博文本情感分类[J]. 计算机应用与软件,2019,36(12):150 - 155.

[15] 宋岩,刘汉永,宁向南,等. 基于层次特征提取的文本分类研究[J]. 计算机应用与软件,2020,37(2):68 - 72,77.

[16] 肖琳,陈博理,黄鑫,等. 基于标签语义注意力的多标签文本分类[J]. 软件学报,2020,31(4):1079 - 1089.

[17] 张璞,陈超,陈韬,等. 两分类器融合的中文微博用户性别分类方法[J]. 计算机工程与设计,2019,40(1):268 - 272.

[18] Fukushima K. Neocognitron: A hierarchical neural network capable of visual pattern recognition[J]. Neural Networks, 1988,1(2):119 - 130.

[19] Lecun Y, Bottou L. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11):2278 - 2324.

(上接第 16 页)

[2] 周晓丹,冯少荣,薛永生. Oracle 10g RAC 核心技术研究与分析[J]. 计算机工程,2007,33(7):53 - 55.

[3] 姜召凤. Oracle RAC 数据库缓存优化方法研究[D]. 大连:大连海事大学,2009.

[4] 杨东日,陈跃,刘姝祎. 面向健康大数据快速读写的存储系统设计[J]. 计算机工程与设计,2018,39(10):3063 - 3067.

[5] 李佳,徐胜超. 基于云计算的智能电网大数据处理平台[J]. 计算机工程与设计,2018,39(10):3073 - 3079.

[6] 赵康,杨余旺. 基于 Hadoop 的物联云监控系统的设计与实现[J]. 计算机与数字工程,2019,47(7):1738 - 1742.

[7] 李斌,郭景维,彭骞. 面向大数据存储的 HBase 二级索引设计[J]. 计算技术与自动化,2019,38(2):124 - 129.

[8] 邱超,许金涛,元晓华. 基于大数据技术的水情云数据中心设计与研究[J]. 浙江大学学报,2019,46(1):92 - 100.

[9] 宋智,徐晓莉,张常亮,等. 应用分布式存储技术优化省级 CIMISS 数据服务能力[J]. 热气象科技,2019,47(3):433 - 438.

[10] 吴燕波,薛琴,向大为,等. 云平台下的 NoSQL 分布式大数据存储技术与应用[J]. 现代电子技术,2016,39(9):44 - 47.

[11] 李波,杜景林,李正方. 基于 SOA 的气象数据共享平台研究[J]. 电子设计工程,2019,27(4):25 - 29.

[12] 曾行吉,任晓炜,宋瑶,等. 微服务在气象数据服务中的应用研究[J]. 气象研究与应用,2019,40(1):80 - 83.

[13] 赵芳,熊安元,张小缨,等. 全国综合气象信息共享平台架构设计技术特征[J]. 应用气象学报,2017,28(6):750 - 757.

[14] 曾行吉,李涛,詹利群,等. 基于 MUSIC 的特色数据与产品回写 CIMISS 方法研究[J]. 气象研究与应用,2018,39(1):111 - 114.

[15] 丁晓刚,鲍广宇,胥秀峰. 一种基于多核并行计算的目标分配算法设计[J]. 指挥控制与仿真,2014,36(5):97 - 99,127.

[16] 李白云,赵春霞. GPU 实时构建四叉树的快速地形渲染算法[J]. 计算机辅助设计与图形学报,2010,22(12):2259 - 2264.