(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 114495114 B
(45)授权公告日 2022.08.05

(21)申请号 202210402975.1

(22)申请日 2022.04.18

(65)同一申请的已公布的文献号
申请公布号 CN 114495114 A

(43)申请公布日 2022.05.13

(73)专利权人 华南理工大学
地址 510000 广东省广州市天河区五山路381号
专利权人 人工智能与数字经济广东省实验室(广州)

(72)发明人 黄双萍 罗钰 徐可可

(74)专利代理机构 东莞卓诚专利代理事务所(普通合伙) 44754
专利代理师 朱鹏

(51)Int.Cl.
G06V 30/19(2022.01)
G06V 30/26(2022.01)
G06K 9/62(2022.01)

(56)对比文件
US 2020402500 A1,2020.12.24
CN 109934293 A,2019.06.25

审查员 张笑迪

权利要求书2页 说明书8页 附图3页

(54)发明名称

基于CTC解码器的文本序列识别模型校准方法

(57)摘要

本发明公开了基于CTC解码器的文本序列识别模型校准方法,包括:将文本图像支撑集输入至待校准训练模型中,获得文本序列识别结果;利用文本图像支撑集的文本序列识别结果计算上下文混淆矩阵,上下文混淆矩阵用于表征序列中相邻时刻预测字符之间的上下文分布关系;根据上下文混淆矩阵,利用上下文相关预测分布对标签平滑中平滑强度有选择性地进行自适应的变化,以实现序列置信度的自适应校准;基于上下文选择性损失函数重新训练待校准训练模型,输出预测文本序列及校准的置信度。本发明方法将标签平滑扩展到基于CTC解码器的文本序列识别模型上,引入序列间上下文关系,对预测序列进行自适应的校准,使得模型输出预测文本置信度能够更加精准。

CN 114495114 B

1.基于CTC解码器的文本序列识别模型校准方法，其特征在于，包括以下步骤：

步骤1，将文本图像支撑集输入至待校准训练模型中，获得文本序列识别结果；

步骤2，利用文本图像支撑集的文本序列识别结果计算上下文混淆矩阵，上下文混淆矩阵用于表征序列中相邻时刻预测字符之间的上下文分布关系；

步骤3，根据上下文混淆矩阵，利用上下文相关预测分布对标签平滑中平滑强度有选择性地进行自适应的变化，以实现序列置信度的自适应校准；

步骤4，基于上下文选择性损失函数重新训练待校准训练模型，最后输出预测文本序列及校准的置信度；

步骤4中所述的上下文选择性损失函数为：

$$\mathrm{Loss}_{CASLS} = -\sum_{t=1}^{T}\sum_{k=1}^{K} \hat{q}_{CASLS}(y_t^k|y_{t-1}^k)\log\, p(y_t^k)$$

$$= \begin{cases} \mathrm{Loss}_{CTC}, & if\ y_t \notin \boldsymbol{E}_{y_{t-1}} \\ (1-\alpha)Loss_{CTC} + \alpha D_{KL}(\frac{C_{y_{t-1},y_t,\hat{y}_t}}{\sum C_{y_{t-1},y_t,j}}||p(y_t)), & if y_t \in \boldsymbol{E}_{y_{t-1}} \end{cases}$$

其中，$\hat{q}_{CASLS}(y_t^k|y_{t-1}^k)$表示$t$时刻时，对应标签类别索引$k$的选择性上下文感知概率分布，$p(y_t^k)$表示在$t$时刻对应预测类别标签$k$的概率，$Loss_{CTC}$表示CTC损失，$D_{KL}$表示$KL$散度，$p(y_t)$表示$t$时刻对应所有$K$类标签类别的概率向量；$\boldsymbol{E}_{y_{t-1}}$表示上一时刻字符$y_{t-1}$所属类别已知时所对应易错集；$T$为预测解码总步长；$K$为预测类别数量；

所述的KL散度定义公式为：

$$\mathrm{D}_{KL}(\mathrm{p}||\mathrm{q}) = -\sum_{k=1}^{K} p(y^k)\log(\frac{p(y^k)}{q(y^k)})$$

$p(y^k)$表示对应预测类别标签k的概率，$q(y^k)$表示对应真实类别标签k的概率。

2.根据权利要求1所述的基于CTC解码器的文本序列识别模型校准方法，其特征在于，所述的计算上下文混淆矩阵的过程，包括以下步骤：

初始化设置共K个预测类别的元素为0的上下文混淆矩阵C_k，k为对应预测类别索引；

比对文本图像支撑集的文本序列识别结果$\hat{Y} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_L\}$和对应真实标签$Y = \{y_1, y_2, \ldots, y_{L'}\}$，$L$为文本序列识别结果长度，$L'$为真实标签序列长度；

若识别结果和真实标签对齐，则在已知上一时刻字符y_{t-1}标签所属类别索引为k情况下，直接统计当前t时刻真实字符y_t被预测为字符$\hat{y}_t$的上下文混淆矩阵C_k，其中，上下文混淆矩阵中每个元素$c_{k,m,n}$表示已知上一时刻真实字符属于第k类时，真实标签属于第m类的当前时刻字符被预测为第n类标签的次数，对于位于文本首位的字符，其上一时刻字符所属类别默认设置为空格；

若识别结果和真实标签未对齐，则先通过编辑距离计算预测序列到真实标签的操作序列，获得序列之间的对齐关系，然后再统计获得上下文混淆矩阵。

3.根据权利要求2所述的基于CTC解码器的文本序列识别模型校准方法，其特征在于，

所述的获得序列之间的对齐关系的过程需要进行若干次下列操作：删除一个字符操作、插入一个字符操作或替换一个字符操作，直到字符正确预测且对齐，其中，删除一个字符操作为纠正真实标签序列中的空符号被错误预测成其他字符，插入一个字符操作为纠正真实标签序列中对应字符被预测成空符号，替换一个字符操作为纠正真实标签序列中对应字符被预测成其他字符。

4.根据权利要求1所述的基于CTC解码器的文本序列识别模型校准方法，其特征在于，所述的易错集的获得，是对K个对应不同预测类别，根据对应上下文混淆矩阵中预测出现在各个类别中的频次，统计得到上一时刻字符y_{t-1}属于第k类类别时的易错集$\boldsymbol{E_k}$，划分依据如下：

$$\boldsymbol{E_k}=\left\{m\left|1-\frac{c_{k,m,m}}{\sum_j c_{k,m,n}}>Th,\mathrm{m}=1,2,\ldots,K\right.\right\}$$

其中，$\frac{c_{k,m,m}}{\sum_j c_{k,m,n}}$表示上一时刻字符$y_{t-1}$属于第$k$类时，当前时刻预测的准确率，若类别错误率大于设定阈值Th，则将对应类别m划入至易错集当中，明确y_{t-1}标签所属类别后，即可获得对应易错集$\boldsymbol{E}_{y_{t-1}}$。

5.根据权利要求1所述的基于CTC解码器的文本序列识别模型校准方法，其特征在于，步骤3中所述的利用上下文相关预测分布对标签平滑中平滑强度有选择性地进行自适应的变化，是指将平滑强度根据上下文关系进行自适应调整，对标签概率进行调整，得到选择性上下文感知概率分布公式为：

$$\hat{q}_{CASLS}(y_t|y_{t-1})=\begin{cases}1,if\hat{y}_t=y_t and y_t\notin \boldsymbol{E}_{y_{t-1}}\\ 1-\alpha\dfrac{1-c_{y_{t-1},y_t,\hat{y}_t}}{\sum c_{y_{t-1},y_t,j}},if\hat{y}_t=y_t and y_t\in \boldsymbol{E}_{y_{t-1}}\\ 0,if\hat{y}_t\neq y_t and y_t\notin \boldsymbol{E}_{y_{t-1}}\\ \alpha\dfrac{c_{y_{t-1},y_t,\hat{y}_t}}{\sum c_{y_{t-1},y_t,j}},if\hat{y}_t\neq y_t and y_t\in \boldsymbol{E}_{y_{t-1}}\end{cases}$$

其中，$c_{y_{t-1},y_t,\hat{y}_t}$表示对于确定上一时刻字符$y_{t-1}$标签所属类别情况下，当前字符$y_t$被预测为$\hat{y}_t$的次数，$j$表示类别索引，$c_{y_{t-1},y_t,j}$表示对于确定上一时刻字符$y_{t-1}$标签所属类别情况下，当前字符$y_t$被预测为$j$的次数，$\alpha$表示平滑强度，对于上一字符$y_{t-1}$标签类别已知情况下，要先确认当前字符$y_t$是否属于易错集，如果不是，则不需要进行标签平滑校准；反之，则根据错误率自适应调整平滑强度。

基于CTC解码器的文本序列识别模型校准方法

技术领域

[0001] 本发明属于人工智能与文本序列处理技术领域，特别是涉及一种基于CTC解码器的文本序列识别模型校准方法。

背景技术

[0002] 随着深度学习的发展，深度神经网络模型因其较高的预测准确度，而在医疗、交通、金融等领域得到了大量的部署，例如：医疗影像识别模型能够为医生诊断病情提供辅助依据，目标检测识别模型让车辆拥有智能分析能力从而控制传感器车速或方向，以及OCR(光学字符识别)模型为金融票据录入数字化提供强力支撑。然而，在深度模型应用在各个领域普及、深化的过程当中，深度模型潜在的风险也逐渐暴露出来。场景文本图像作为我们日常场景中广泛存在的数据形式之一，广泛存在于我们生活的各个行业、领域当中。例如：医疗诊断中的问诊记录、医疗检查单和金融系统中票据等文本数据。相比于普通单帧图像、字符这种非结构化数据，结构化的序列数据预测更加困难，其可靠性的获取和判断也更加复杂。

[0003] 目前，置信度是评价预测可靠性最直接的指标之一。一般将模型预测分数经过归一化为概率后作为其置信度。可靠的置信度能够确切地反映预测的准确度，当模型对于预测并不自信并且给出一个相对低的置信度时，人工需要介入决策，以保证任务的安全进行。然而，研究发现，现有的许多深度神经网络模型输出的置信度并不校准，而是存在过自信的问题，即输出的置信度要高于准确度。模型不校准的原因来自多个方面。一方面，随着模型本身的结构日渐庞大、复杂，大量参数带来的高拟合能力，导致模型出现过拟合的问题。对于过拟合的预测标签类别，模型倾向给错误的预测也分配一个高置信度。而且，基于one-hot编码的损失函数和softmax置信度的计算方式加大了正负预测样本之间的距离，尽管这样便于挑选出正确样本，但也容易导致预测置信度过于自信。另一方面，训练数据和测试数据的分布差异。当模型需要处理现实场景中从未或很少在训练数据集中见到的数据，模型也很难给出一个可靠的置信度。

[0004] 而由于文本序列的复杂结构，针对场景文本识别模型的校准也十分困难。具体而言，一是文本序列通常由多个字符组成，其置信度空间大小也会随着字符的数量增加的变大。二是文本识别通常是一个时序相关的过程，字符之间的上下文关系是一个重要的先验信息。而且对于不同的字符，它们之间的上下文依赖强度也是不同的。因此，序列级的校准很难通过简单地对所有字符进行统一的校准实现。

[0005] 然而，现有大部分置信度校准方法主要是针对非结构化的简单数据展开。这些校准方法主要可以分为两大类：后处理校准和预测模型训练校准。后处理的方式一般在留出(held-out)数据集上学习置信度相关的回归方程，对输出置信度进行变换。早期在机器学习领域提出的面向传统分类器的校准方法大多是基于后处理的思想，例如：Plattscaling、保序回归、histogrambinning等。而在深度学习领域，有学者在plattscaling基础上提出Temperaturescaling，通过引入温度参数对置信度进行校准。预测模型训练校准则一般直

接对深度模型进行调整。这一类方法主要考虑到过拟合带来的过自信问题，通过dropout、标签平滑损失、熵正则等方式，通过缓解过拟合以校准模型。另外，从数据层面考虑，部分方法则是在训练过程中对训练数据进行增强操作来解决这一问题，例如MixUp、GAN等方法。但是，这些方法没有考虑到数据集中不同类别数据分布的不均匀性，或者只考虑了局部的单个预测和真实标签之间的相关性，忽略了序列数据的长度和内在上下文依赖特性，都很难直接迁移到序列数据的置信度校准上。因此，需要进一步根据序列数据特性做出针对性的校准设计，提升序列置信度的校准性能。

发明内容

[0006] 有鉴于此，有必要针对场景文本识别模型的置信度校准的技术问题，提供一种基于CTC解码器的文本序列识别模型校准方法，所述方法重新审视标签平滑方法的本质，标签平滑的有效性主要通过在原始损失函数基础上添加Kullback-Leibler (KL)散度项作为正则项体现出来，同时考虑到序列中存在的上下文依赖，以混淆矩阵的形式对字符之间的上下文关系进行建模，将其作为语言知识先验指导标签概率分布，不同类别标签的平滑强度将根据其上下文预测错误率进行自适应的调整。

[0007] 本发明公开了基于CTC解码器的文本序列识别模型校准方法，包括以下步骤：

[0008] 步骤1，将文本图像支撑集输入至待校准训练模型中，获得文本序列识别结果；

[0009] 步骤2，利用文本图像支撑集的文本序列识别结果计算上下文混淆矩阵，上下文混淆矩阵用于表征序列中相邻时刻预测字符之间的上下文分布关系；

[0010] 步骤3，根据上下文混淆矩阵，利用上下文相关预测分布对标签平滑中平滑强度有选择性地进行自适应的变化，以实现序列置信度的自适应校准；

[0011] 步骤4，基于上下文选择性损失函数重新训练待校准训练模型，最后输出预测文本序列及校准的置信度。

[0012] 具体地，所述的计算上下文混淆矩阵的过程，包括以下步骤：

[0013] 初始化设置共K个预测类别的元素为0的上下文混淆矩阵C_k，k为对应预测类别索引；

[0014] 比对文本图像支撑集的文本序列识别结果$\hat{Y}=\{\hat{y}_1,\hat{y}_2,\dots,\hat{y}_L\}$和对应真实标签$Y=\{y_1,y_2,\dots,y_{L'}\}$，$L$为文本序列识别结果长度，$L'$为真实标签序列长度；

[0015] 若识别结果和真实标签对齐，则在已知上一时刻字符y_{t-1}标签所属类别索引为k情况下，直接统计当前t时刻字符y_t被预测为字符$\hat{y}_t$的上下文混淆矩阵C_k，其中，上下文混淆矩阵中每个元素$c_{k,m,n}$表示已知上一时刻预测字符属于第k类时，真实标签属于第m类的当前时刻字符被预测为第n类标签的次数，对于位于文本首位的字符，其上一时刻字符所属类别默认设置为空格；

[0016] 若识别结果和真实标签未对齐，则先通过编辑距离计算预测序列到真实标签的操作序列，获得序列之间的对齐关系，然后再统计获得上下文混淆矩阵。

[0017] 优选地，所述的获得序列之间的对齐关系的过程需要进行若干次下列操作：删除一个字符操作、插入一个字符操作或替换一个字符操作，直到字符正确预测且对齐，其中，删除一个字符操作为纠正真实标签序列中的空符号被错误预测成其他字符，插入一个字符

操作为纠正真实标签序列中对应字符被预测成空符号，替换一个字符操作为纠正真实标签序列中对应字符被预测成其他字符。

[0018] 具体地，步骤3中所述的利用上下文相关预测分布对标签平滑中平滑强度有选择性地进行自适应的变化，是指将平滑强度根据上下文关系进行自适应调整，对标签概率进行调整，得到选择性上下文感知概率分布公式为：

[0019] $$\hat{q}_{CASLS}(y_t|y_{t-1})=\begin{cases}1, if\ \hat{y}_t=y_t and y_t\notin \boldsymbol{E}_{y_{t-1}}\\ 1-\alpha\dfrac{1-c_{y_{t-1},y_t,\hat{y}_t}}{\sum c_{y_{t-1},y_t,j}}, if\ \hat{y}_t=y_t and y_t\in \boldsymbol{E}_{y_{t-1}}\\ 0, if\ \hat{y}_t\neq y_t and y_t\notin \boldsymbol{E}_{y_{t-1}}\\ \alpha\dfrac{c_{y_{t-1},y_t,\hat{y}_t}}{\sum c_{y_{t-1},y_t,j}}, if\ \hat{y}_t\neq y_t and y_t\in \boldsymbol{E}_{y_{t-1}}\end{cases}$$

[0020] 其中$\boldsymbol{E}_{y_{t-1}}$，表示上一时刻字符y_{t-1}所属类别已知时所对应易错集，$c_{y_{t-1},y_t,\hat{y}_t}$表示对于确定上一时刻字符$y_{t-1}$标签所属类别情况下，当前字符$y_t$被预测为$\hat{y}_t$的次数，$j$表示类别索引，$c_{y_{t-1},y_t,j}$表示对于确定上一时刻字符$y_{t-1}$标签所属类别情况下，当前字符$y_t$被预测为$j$的次数，$\alpha$表示平滑强度，对于上一字符$y_{t-1}$标签类别已知情况下，要先确认当前字符$y_t$是否属于易错集，如果不是，则不需要进行标签平滑预测；反之，则根据错误率自适应调整平滑强度；

[0021] 所述的易错集的获得，是对K个对应不同预测类别，根据对应上下文混淆矩阵中预测出现在各个类别中的频次，统计得到上一时刻字符y_{t-1}属于第k类时的易错集$\boldsymbol{E}_k$，划分依据如下：

[0022] $$\boldsymbol{E}_k=\left\{m\middle|1-\frac{c_{k,m,m}}{\sum_j c_{k,m,n}}>Th, \mathrm{m}=1,2,\dots,K\right\}$$

[0023] 其中，$\frac{c_{k,m,m}}{\sum_j c_{k,m,n}}$表示上一时刻字符$y_{t-1}$属于第$k$类时，当前时刻预测的准确率，若类别错误率大于设定阈值Th，则将对应类别m划入至易错集当中，明确y_{t-1}标签所属类别后，即可获得对应易错集$\boldsymbol{E}_{y_{t-1}}$。

[0024] 更具体地，步骤4中所述的上下文选择性损失函数为：

$$\mathrm{Loss}_{CASLS}=-\sum_{t=1}^{T}\sum_{k=1}^{K}\hat{q}_{CASLS}(y_t^k|y_{t-1}^k)\log p(y_t^k)$$

[0025]

$$=\begin{cases}\mathrm{Loss}_{CTC}, & if\ y_t\notin \boldsymbol{E}_{y_{t-1}}\\ (1-\alpha)Loss_{CTC}+\alpha D_{KL}(\dfrac{c_{y_{t-1},y_t,\hat{y}_t}}{\sum c_{y_{t-1},y_t,j}}||p(y_t)), & if\ y_t\in \boldsymbol{E}_{y_{t-1}}\end{cases}$$

[0026] 其中，$\hat{q}_{CASLS}(y_t^k|y_{t-1}^k)$表示$t$时刻时，对应标签类别索引$k$的选择性上下文感知概率分布，$p(y_t^k)$表示在$t$时刻对应预测类别标签$k$的概率，$Loss_{CTC}$表示CTC损失，$D_{KL}$表

示KL散度，$p(y_t)$表示t时刻对应所有K类标签类别的概率向量，

[0027] KL散度定义公式为：

[0028] $$D_{KL}(p||q) = -\sum_{k=1}^{K} p(y^k)\log\left(\frac{p(y^k)}{q(y^k)}\right)$$

[0029] $p(y^k)$表示对应预测类别标签k的概率，$q(y^k)$表示对应真实类别标签k的概率。

[0030] 与现有技术相比，本发明的有益效果在于：

[0031] 本发明方法将标签平滑扩展到基于CTC解码器的文本序列识别模型上，同时引入序列间上下文关系，对预测序列进行自适应的校准，能够很好改善文本序列识别模型的校准性能，使得模型输出预测文本置信度能够更加精准。

附图说明

[0032] 图1示出了本发明实施方法的流程示意图；

[0033] 图2示出了本发明实施例各模块工作的结构示意图；

[0034] 图3示出了本发明实施例中对齐策略的示意流程。

具体实施方式

[0035] 为使本发明实施例的目的、技术方案和优点更加清楚，下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本发明保护的范围。

[0036] 为了引用和清楚起见，下文中使用的技术名词、简写或缩写总结解释如下：

[0037] CTC:Connectionist Temporal Classification（连结主义时序分类器）

[0038] KL散度:Kullback-Leibler散度

[0039] NLL:Negative Log-Likelihood（负对数似然）

[0040] LS:Label Smoothing（标签平滑）

[0041] CASLS:Context-aware Selective Label Smoothing(上下文感知选择性标签平滑)

[0042] 本发明公开了一种基于形状变换的字符图像增广方法，以解决现有技术中存在的诸多问题。

[0043] 图1示出了本发明实施例的流程示意图。基于CTC解码器的文本序列识别模型校准方法，包括以下步骤：

[0044] 将文本图像支撑集输入至待校准训练模型中，获得文本序列识别结果；

[0045] 利用文本图像支撑集的文本序列识别结果计算上下文混淆矩阵，上下文混淆矩阵用于表征序列中相邻时刻预测字符之间的上下文分布关系；

[0046] 根据上下文混淆矩阵，利用上下文相关预测分布对标签平滑中平滑强度有选择性地进行自适应的变化，以实现序列置信度的自适应校准；

[0047] 基于上下文选择性损失函数重新训练待校准训练模型，最后输出预测文本序列及校准的置信度。

[0048] 具体地，本实施例采用如下步骤进行发明方法的实施。

[0049] 步骤1、构建支撑数据集，将其输入至对应场景文本识别预训练模型中，获得识别结果，即相应文本序列。

[0050] 支撑集中数据分布需要与训练集相似，一般选择基准数据集的验证集或者部分训练集作为支撑集。在这里，选择IIIT5k、SVT、IC03、IC13以及IC15 的训练数据集集合作为支撑集。将待测数据输入至对应场景文本识别预训练模型后，经过模型识别预测，得到对应预测序列。于下一步混淆矩阵构建。

[0051] 步骤2、利用支撑集预测结果获取序列上下文预测分布关系，并将其以混淆矩阵的形式表示，作为上下文建模输出。

[0052] 在步骤2中，对于输入数据，在预测文本$\hat{Y}=\{\hat{y}_1,\hat{y}_2,\dots,\hat{y}_L\}$和对应真实标签$Y=\{y_1,y_2,\dots,y_{L'}\}$基础上（$L$和$L'$为对应文本序列长度），获取序列中相邻时刻预测的上下文关系，即在已知上一时刻字符预测所属类别情况下，其下一时刻字符预测属于类别的概率与其有一定相关性。

[0053] 若识别结果和真实标签对齐，则在已知上一时刻字符y_{t-1}标签所属类别索引为k情况下，直接统计当前t时刻字符y_t被预测为字符$\hat{y}_t$的上下文混淆矩阵C_k。其中，上下文混淆矩阵中每个元素$c_{k,m,n}$元示已知上一时刻预测字符属于第k类时，真实标签属于第m类的当前时刻字符被预测为第n类标签的次数。特殊地，对于位于文本首位的字符，其上一时刻字符所属类别默认设置为空格；

[0054] 具体构建方式为，首先初始化设置共K个预测类别的元素为0的混淆矩阵。对于场景文本识别，首先初始化$K=37$（包含10个数字、26个英文字母以及1个空格类别）的混淆矩阵。假设有预测序列“cat”（真实标签为“cat”），对首字符“c”上一时刻字符类别空格，当前时刻真实字符标签“c”被正确预测为“c”，则在对应空格类别混淆矩阵中表示相应标签“c”和预测“c”的位置元素上加一。以同样的方式对支撑集中所有样本进行统计，最终得到表示不同预测类别上下文预测频次分布的混淆矩阵。附图2分别表示上一时刻字符为“3”，“A”，“V”的上下文混淆矩阵，对于不同类别上一时刻字符预测，当前字符预测所属类别也不同，因此需进行差异化校准操作。

[0055] 其中，考虑到预测序列因错误预测而导致和其真实序列标签不对齐的情况，使用编辑距离计算真实序列和预测序列之间的操作序列，获得序列之间的对齐关系。具体对齐策略如附图3所示，为使预测文本和真实文本实现字符之间的一一对应，需要进行(1)删除一个字符(d)；(2)插入一个字符(i)；(3)替换一个字符(s)的操作。若无操作，用符号“-”表示。以预测序列“lapaitmen”为例，为获取其到真实标签“apartment”的编辑距离，需经过如下操作：删除字符“l”、替换字符“i”为“r”、插入字符“t”。相应地，在统计混淆矩阵过程中，删除操作表示真实序列中的空符号标签“#”被错误预测成其他字符“l”，插入操作表示真实序列中对应标签“t”被预测成空符号“#”。

[0056] 步骤3、利用混淆矩阵，将标签平滑根据上下文关系进行自适应的变化，引入惩罚项以实现序列置信度的自适应校准。

[0057] 在步骤3中，对于CTC解码器而言，其优化目标为序列概率的最大似然，其定义可由下式表示：

[0058]
$$\begin{aligned} Loss_{CTC} &= -lnP(l|x) \\ &= -ln \sum_{\pi \in B^{-1}(l)} p(\pi|x) \\ &= -ln \sum_{\pi \in B^{-1}(l)} \prod_{t=1}^{T} y_{\pi}^{t} \end{aligned}$$

[0059] 其中，$P(l|x)$为给定输入x，输出为l的概率，T为预测解码总步长，y_{π}^{t}表示在解码路径π中第t时刻所对应字符的置信度，B表示映射规则。该解码路径的概率被直接视作为预测序列的置信度。

[0060] 然后，将一般用于交叉熵损失上的标签平滑策略推广到CTC损失上。对标签平滑损失进行推导。标签平滑概率分布可表示为：

[0061]
$$\hat{q}(y_t) = (1-\alpha)q(y_t) + \alpha U(y_t)$$

[0062] 其中，$\hat{q}(y_t)$为均匀标签平滑概率，α为平滑因子，$q(y_t)$标签符合one-hot概率分布，若预测正确，则为1，否则为0，相当于一个狄拉克函数，而$U(y_t)$标签概率在所有类别标签上的均匀分布，值$1/K$为。将上式代入一般文本序列识别损失函数中可得：

[0063]
$$\begin{aligned} \text{Loss}_{LS} &= -\sum_{t=1}^{T}\sum_{k=1}^{K} \hat{q}(y_t^k)\log p(y_t^k) \\ &= -\sum_{t=1}^{T}\sum_{k=1}^{K} \left((1-\alpha)q(y_t^k) + \alpha U(y_t^k)\right)\log p(y_t^k) \\ &= -(1-\alpha)\sum_{t=1}^{T}\sum_{k=1}^{K} q(y_t^k)\log p(y_t^k) - \alpha\sum_{t=1}^{T}\sum_{k=1}^{K} U(y_t^k)\log p(y_t^k) \end{aligned}$$

[0064] 其中，$p\left(y_t^k\right)$为t时刻预测为k类别的概率，T为预测解码总步长。

[0065] 根据KL散度定义：

[0066]
$$\text{D}_{KL}(\text{p}||\text{q}) = -\sum_{k=1}^{K} p(y^k)\log\left(\frac{p(y^k)}{q(y^k)}\right)$$

[0067] 损失函数可以被推导解耦为标准negative log likelihood(NLL)损失和KL散度项D_{KL}两项：

[0068]
$$\begin{aligned} \text{Loss}_{LS} &= -(1-\alpha)\sum_{t=1}^{T}\sum_{k=1}^{K} q(y_t^k)\log p(y_t^k) - \alpha\sum_{t=1}^{T}\sum_{k=1}^{K} U(y_t^k)\log p(y_t^k) \\ &= (1-\alpha)\text{Loss}_{NLL} + \alpha\sum_{t=1}^{T} D_{KL}\left(U(y_t) \parallel p(y_t)\right) + \delta \end{aligned}$$

[0069] 其中，δ表示一个常数项，在梯度反传中没有影响，可忽略。

[0070] 由于函数Loss_{LS}期望整体损失趋近于零，对于KL散度惩罚项，可近似理解为期望预测概率分布和均匀分布距离也越小，防止预测概率朝着过自信的方向变化。因此，尽管基于CTC解码器的文本序列识别模型不对真实标签进行one-hot编码，但其核心优化目标依然为序列置信度最大化，则结合标准标签平滑CTC损失可被定义为：

[0071] $$\mathrm{Loss}_{LS} = (1-\alpha)\mathrm{Loss}_{CTC} + \alpha \sum_{t=1}^{T} D_{KL}\left(U(y_t) \parallel p(y_t)\right)$$

[0072] 其中，Loss_{CTC}为CTC损失。平滑因子α作为惩罚项的权重，控制校准的强度。实施中设置$\alpha = 0.05$。

[0073] 进一步在标签平滑中引入序列上下文关系。首先筛选易错集，只对易错类别进行标签平滑。对K个对应不同预测类别，可以根据对应混淆矩阵中预测出现在各个类别中的频次，统计得到上一时刻字符y_{t-1}属于第k类时的易错集$\boldsymbol{E_k}$。其划分依据如下：

[0074] $$\boldsymbol{E_k} = \left\{ m \middle| 1 - \frac{c_{k,m,m}}{\sum_j c_{k,m,n}} > Th, \mathrm{m} = 1,2,\ldots,K \right\}$$
#

[0075] 其中，$\frac{c_{k,m,m}}{\sum_j c_{k,m,n}}$表示上一时刻字符$y_{t-1}$属于第$k$类时，当前时刻预测的准确率。若类别错误率大于设定阈值Th，则将对应类别划入至易错集当中，明确y_{t-1}标签所属类别后，即可获得对应易错集$\boldsymbol{E}_{\boldsymbol{y_{t-1}}}$。实施中设置$T = 0.05$。

[0076] 根据步骤2中获得的混淆矩阵，进一步将平滑强度根据上下文关系进行自适应调整。对标签概率进行调整，得到选择性上下文感知(CASLS)概率分布公式：

[0077] $$\hat{q}_{CASLS}(y_t|y_{t-1}) = \begin{cases} 1, if\, \hat{y}_t = y_t and y_t \notin \boldsymbol{E}_{\boldsymbol{y_{t-1}}} \\ 1 - \alpha \dfrac{1 - c_{y_{t-1},y_t,\hat{y}_t}}{\sum c_{y_{t-1},y_t,j}}, if\, \hat{y}_t = y_t and y_t \in \boldsymbol{E}_{\boldsymbol{y_{t-1}}} \\ 0, if\, \hat{y}_t \neq y_t and y_t \notin \boldsymbol{E}_{\boldsymbol{y_{t-1}}} \\ \alpha \dfrac{c_{y_{t-1},y_t,\hat{y}_t}}{\sum c_{y_{t-1},y_t,j}}, if\, \hat{y}_t \neq y_t and y_t \in \boldsymbol{E}_{\boldsymbol{y_{t-1}}} \end{cases}$$

[0078] 对于上一字符y_{t-1}标签类别已知情况下，要先确认当前字符y_t是否属于易错集，如果不是，则不需要进行标签平滑预测；反之，则根据错误率自适应调整平滑强度。

[0079] 将该概率分布带入至标签平滑CTC损失中，可得基于CTC解码器的选择性上下文感知损失：

$$\mathrm{Loss}_{CASLS} = -\sum_{t=1}^{T}\sum_{k=1}^{K} \hat{q}_{CASLS}(y_t^k|y_{t-1}^k)\log p(y_t^k)$$

[0080]

$$= \begin{cases} \mathrm{Loss}_{CTC}, & if\ y_t \notin \boldsymbol{E}_{\boldsymbol{y_{t-1}}} \\ (1-\alpha)Loss_{CTC} + \alpha D_{KL}(\dfrac{c_{y_{t-1},y_t,\hat{y}_t}}{\sum c_{y_{t-1},y_t,j}} || p(y_t)), & if y_t \in \boldsymbol{E}_{\boldsymbol{y_{t-1}}} \end{cases}$$

[0081] 考虑到计算KL散度时，输出概率是预测路径的概率，存在其长度和真实标签长度存在不对齐的情况，对于预测概率$p(y_t)$，只保留预测路径映射后序列所在的位置。然后，根据步骤2中所述编辑距离对齐策略，对于删除操作，在对应目标序列空白位置添加空格的one-hot编码；对于插入操作，在预测序列对应的空白位置添加均分分布概率向量；而替换操作不会给概率分布带来任何变化。

[0082] 步骤4、调整损失函数后重新训练目标模型，最后输出预测序列及其校准置信度。

[0083] 在步骤4中，根据步骤3中上下文感知选择性标签平滑策略对原始损失函数进行调整，重新训练目标过置信模型，使模型能够得到校准。由于采用微调模型进行训练，训练过程中设置学习率为10^{-5}，迭代训练200000个次后，最终输出预测文本及其校准后的置信度。

[0084] 本领域普通技术人员可以意识到，结合本文中所公开的实施例描述的各示例的单元及算法步骤，能够以电子硬件、或者计算机软件和电子硬件的结合来实现。这些功能究竟以硬件还是软件方式来执行，取决于技术方案的特定应用和设计约束条件。专业技术人员可以对每个特定的应用来使用不同方法来实现所描述的功能，但是这种实现不应认为超出本发明的范围。

[0085] 以上实施例的各技术特征可以进行任意的组合，为使描述简洁，未对上述实施例中的各个技术特征所有可能的组合都进行描述，然而，只要这些技术特征的组合不存在矛盾，都应当认为是本说明书记载的范围。

[0086] 本领域技术人员应明白，本申请的实施例可提供为方法、系统或计算机程序产品。因此，本申请可采用完全硬件实施例、完全软件实施例或结合软件和硬件方面的实施例的形式。而且，本申请可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质(包括但不限于磁盘存储器、CD-ROM、光学存储器等)上实施的计算机程序产品的形式。

[0087] 以上所述仅是本发明的优选实施方式，应当指出，对于本技术领域的普通技术人员来说，在不脱离本发明原理的前提下，还可以做出若干改进和润饰，这些改进和润饰也应视为本发明的保护范围。

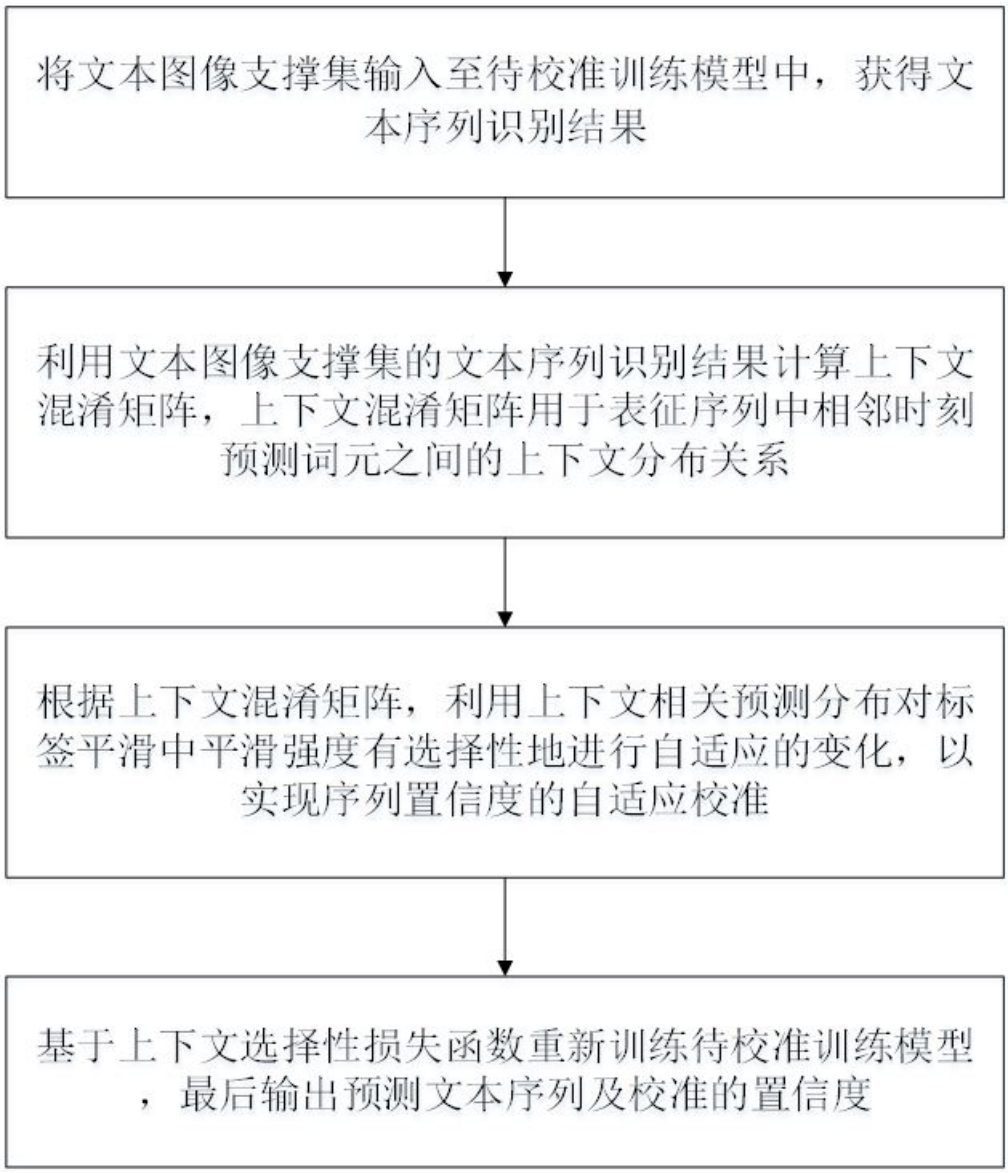
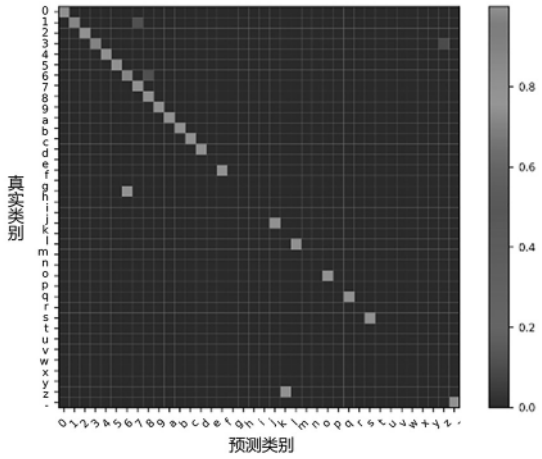
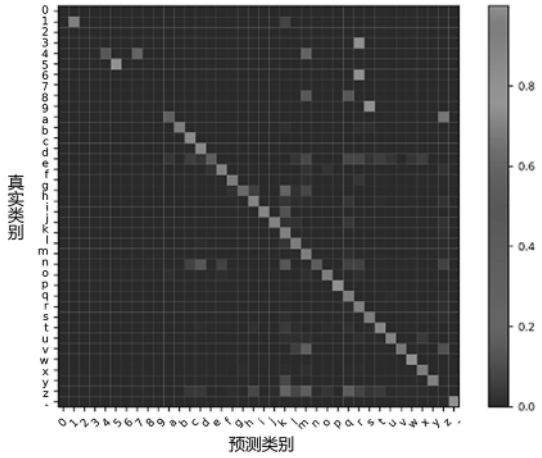

将文本图像支撑集输入至待校准训练模型中，获得文本序列识别结果
利用文本图像支撑集的文本序列识别结果计算上下文混淆矩阵，上下文混淆矩阵用于表征序列中相邻时刻预测词元之间的上下文分布关系
根据上下文混淆矩阵，利用上下文相关预测分布对标签平滑中平滑强度有选择性地进行自适应的变化，以实现序列置信度的自适应校准
基于上下文选择性损失函数重新训练待校准训练模型，最后输出预测文本序列及校准的置信度

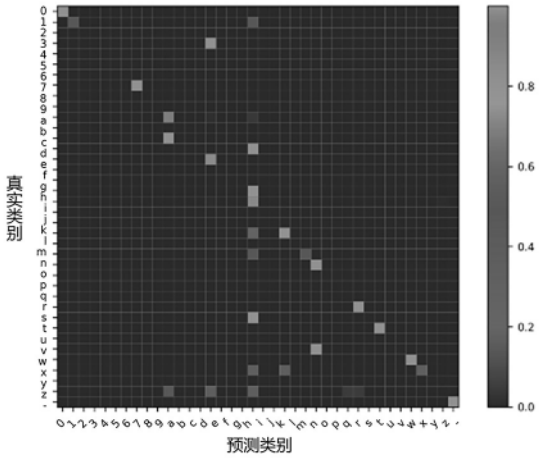
图1



字符“3”的上下文混淆矩阵



字符“A”的上下文混淆矩阵



字符“V”的上下文混淆矩阵

图2

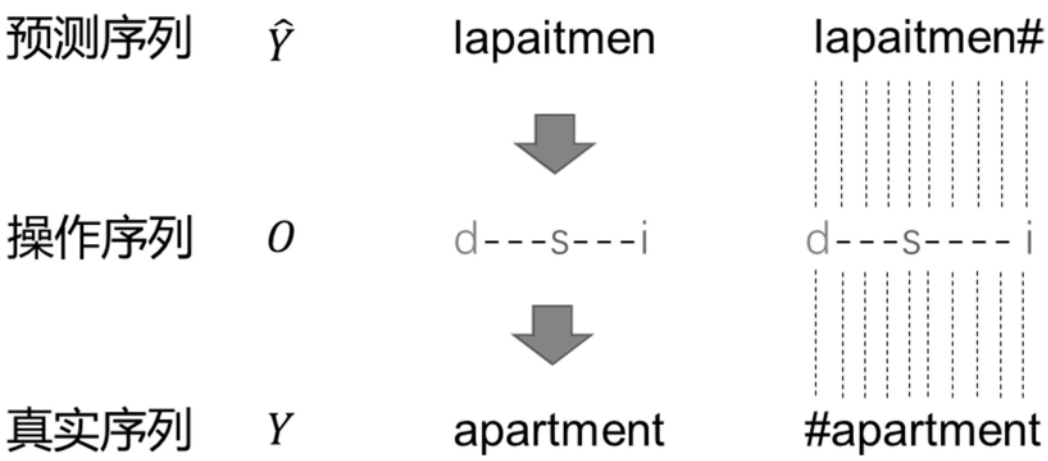

预测序列
$\hat{Y}$
lapaitmen
lapaitmen#
操作序列
O
d---s---i
d---s----i
真实序列
Y
apartment
#apartment

图3