通过自适应改进书籍OCR

语言和图像模型

李达祥
Google Inc.
dsl@google.com

雷·史密斯Goo
gle Inc.
Rays@google.com

摘要—为了应对书籍内容和字体的多样性，OCR系统必须利用书籍中的强一致性，并适应不同书籍的变化，这一点很重要。在这项工作中，我们描述了一个系统，该系统使用特定于文档的图像和语言模型将两个并行校正路径组合在一起。每种模型都适应书中的形状和词汇，以将不一致之处识别为校正假设，但依靠另一种模型进行有效的交叉验证。使用开源Tesseract引擎作为基准，在大量扫描书籍数据上的结果表明，使用此方法可以将字错误率降低25％。

关键字：特定于文档的OCR；自适应OCR；错误更正

一，引言

数字图书馆的到来激发了研究对构建可适应多种内容和字体的自适应OCR系统的兴趣。除了对不断增长的数据量进行培训外，人们普遍认为，可以利用大型扫描集合中强大的冗余性和一致性来提高系统的准确性和鲁棒性。

书籍自适应OCR与说话者有很多相似之处语音识别中的适应[5]。通过适当选择分类器体系结构，可以使用EM [7,10]在最大似然或MAP标准下针对未标记的测试样本调整与基本书无关的模型的参数。但是，该方法通常不适用于任意分类器。

现代OCR引擎通常采用动态学习通过选择静态分类器产生的一组可靠决策进行再训练的方法[8]。通过选择可靠的单词来重新训练特殊的分类器[2]，也可以将其作为后处理来进行。这种方法主要取决于选择正确的单词进行改编的成功。

一些研究从没有任何静态开始就进一步发展了形状分类器，并试图通过探索形状簇并通过语言模型[1,3]迭代传播标签分配来自动解码文本。这些解密解决方案非常适合处理任何训练样本都没有的稀有字体，但不足以独自实现高精度性能。

以上方法可以表征为基于图像的全局语言模型驱动的更正。精度这些基于图像的自适应方法取决于基础语言模型的适当性。有限的研究同时修改了书籍中的图像和语言模型。 Xiu＆Baird [11]最小化基于图像的图标模型和基于单词的语言模型之间的全局互熵，以优化整本书的识别能力。不幸的是，这样的全局优化策略收敛缓慢。

在这项工作中，我们将基于图像的自适应这种方法和基于语言的校正可以独立地检测模型中的不一致之处以进行校正，但是要依靠其他模型的交叉验证进行验证。这是由于以下事实引起的：许多形状相似的混淆（例如m / rn和1 / l / I）难以在所有字体之间可靠地分开，但通常容易与上下文区分开。另一方面，基于语言的校正在从图像中容易区分的“黑色”和“蓝色”之间进行选择时将无效。目标是利用每种模型的优势来解决最明显的混淆。此外，我们允许图像模型和语言模型都适应文档，从而对内容和字体多样性更加健壮。我们证明，与基线开源OCR引擎Tesseract [8]相比，与文档相关的语言和图像校正可以串联工作得到显着改善。

二。系统总览

A.设计动机我们的系统设计受到几个关键因素的激励。

首先，观察到大多数混淆仅在图像或语言上是模棱两可的，这表明交叉模型验证将解决最明显的情况。我们没有在图像和语言模型的一致性和复杂性上同时优化全局成本函数，而是采用了一种更简单但更有效的方法来让每个校正路径独立运行，并通过交叉验证松散耦合。

其次，虽然大多数研究都集中在形状适应上为了应对由全局静态语言模型驱动的字体样式和扫描特征的变化，我们注意到文档词汇差异很大，并且由于基本分布的先验概率较低，因此主题关键字经常被不公平地降级。为了处理书籍内容和字体的多样性，图像模型和语言模型都必须是自适应的。

与所有自适应系统一样，问题是
决定哪些答案足够可靠以适应，以及如何使用该信息来改进模型。接下来描述总体控制策略和系统组件。

B.系统控制我们的系统由两条相似的校正路径组成，

一种基于图像和一种基于语言的方法，如图1所示。在前者中，形状相似的碎片被认为是同一类。因此，形状看起来与带有不同标签的聚类相比看起来与具有相同标签的聚类更相似。可以通过咨询其他形状相似的群集来解决冲突。形状簇和标签在扩展到更大的上下文以包括其相邻符号时也应保持一致。

类似的方法可以应用于基于语言的
更正。拼写相同且独立于周围标点符号和表面形式的单词标记被视为同一类。在上下文中所有出现的情况下评估此令牌时，应具有最高的可能性。语音以外的令牌被认为是可疑的，并在所有实例中进行评估，以查找是否存在更可能的答案。在歧义的语言环境中，这可能会产生有效的拼写更正，还会产生错误的假设。图像模型会评估每个校正实例，以拒绝那些明显的不匹配。

验证后，接受的更改将在
输出文件和文档模型。语言模型将获得一个新的经过验证的单词列表，而图像模型则需要更新形状簇。进行的特定更正也会累积起来，以跟踪常见的混乱情况。显然，此过程本质上是迭代的，确切序列可能有许多变体。我们发现确切的操作顺序（例如，两个模型上所有更改的同步更新）并没有太大的区别。另外，尽管我们能够通过更多的迭代获得更多的改进，但是大多数性能提升是在第一次迭代中实现的。

C.系统组件图像模型由静态（基本）形状组成

模型和动态（自适应）模型。基本形状模型通常在标签数据上针对大量字体进行训练，并用作所有书籍的基线，而自适应模型则依赖于无监督学习来构造特定于文档的模板。为了便于讨论，基本模型是OCR引擎，可以调用该引擎进行一般形状分类。自适应模型是通过对字符形状进行聚类构建的，这些字符形状由基本模型分配的标签分组。通过探索聚类结构并将字符与相邻字符进行比较，我们可以识别标签和形状之间的不一致。如前所述，在进行这种类型的改编以纠正标签方面已有很多研究，但是我们的形状模型具有利用双字母组的形状以及单个字符的形状的独特功能，这有助于纠正由原始字符造成的分割错误OCR引擎。我们依靠语言模型来验证是否需要更正。

语言模型类似于图像模型中的
很多方法。它还包括一个静态基础模型和一个
自适应文档（缓存）模型。基本模型针对语言（或子集合）的所有数据进行训练，而高速缓存模型是由OCR产生的可疑标签构造的。使用缓存模型的原因是词汇量在书本之间差异很大。单一的基本模型通常会不公平地惩罚书中出现频率很高的特定主题词。在我们的案例中，基本模型是一个大词n-gram模型，其后退到词汇外（OOV）词的字符n-gram模型。高速缓存模型由高置信度单词和已由基本语言模型和图像模型验证的单词构成。对于每个可疑单词，我们将在上下文中共同评估所有出现的单词，以识别潜在的更正。这些假设然后由图像模型验证，以确定它们是否被接受。

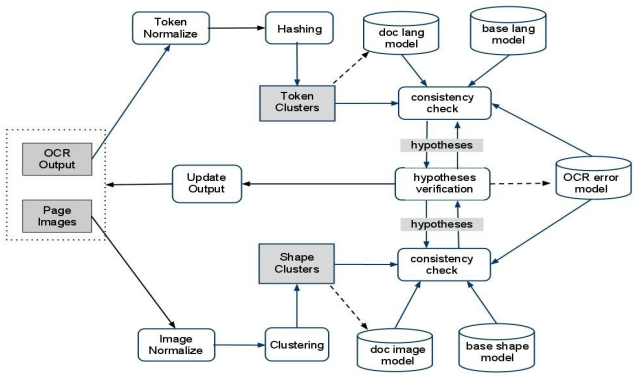

图1.基于两个并行校正路径的建议系统

特定于文档的图像和语言模型。

三，基于图像的校正

自适应图像模型的构建分为两个阶段：
形状聚类和形状分类。

A.形状聚类首先，使用类提取单个组件的形状

OCR引擎生成的标签和边界框。 OCR引擎识别的每个单词图像都经过处理，以通过自动反转以使文本变黑来增强灰度图像，并通过增强对比度来使用整个8位范围。使用OCR的边界框将单独的和相邻的字符对（剪辑）从清理的单词图像中切出，以进行聚类。

每个类的标签和大小（在几个像素之内）为
由形状聚类单独处理。使用剪辑中的像素位置作为尺寸来构造特征向量。由于片段的大小略有不同，因此所有像素位置都是相对于片段的质心计算的。使用基于kd树的层次聚类聚类（KDHAC）对片段进行聚类。 KDHAC在树的每个级别上枢转输入特征向量的不同维度，但是在此应用程序中，维度的数量远远超过了kd-tree中所需的级别的数量，因此，通过减小维度来对维度进行排序为了减少计算时间，将非常接近现有树节点的剪辑保留在该节点上，而不将其压入树中

形状聚类和形状分类都需要一个

两个图像或两个聚类平均值之间的距离度量。该度量标准基于模板匹配，但是由于笔触的边缘可能未完全对齐，因此灰度差异平方的简单总和是不够的，但是希望将“ i”与“í”区分开。距离量度惩罚了在具有低梯度且具有接近黑色或白色的灰度值之一的区域中发生的差异。

B.形状分类所有群集最初都标记为Master类型。所有集群

然后将它们与具有更多样本的相同类别标签的所有其他聚类进行比较，如果距离小于阈值，则将较小聚类的类型更改为从属（从属）。然后，从属集群遵循其分配的主节点的命运。

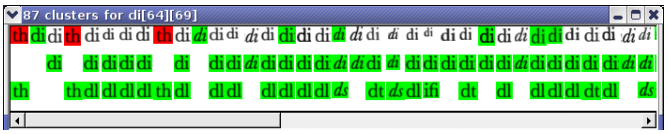

图2.形状簇校准的结果。主群集，拒绝群集和从属群集分别被着色为绿色，红色和透明。

现在将每个主群集与每个主群集进行比较

具有更多样本，但具有不同的类别标签。如果距离小于阈值，则将较小的群集重新键入为Reject，并将进行可能的更正。图2显示了此操作的结果。第一行显示的是按频率排序的“ di”类群集，绿色表示“主”，红色表示“拒绝”，无颜色表示“从属”。第二行显示的是紧邻上方的群集的“ di”的最接近的Master，而第三行显示的是除“ di”以外的最近的Class的Master。实际上是“ th”的三个“ di”的群集被标记为拒绝。

请注意包含二元组，大多数的更正

无需重新考虑字符分割就可以实现分割错误。同样，由于与另一个二元组的匹配，这种“ di”-> th校正在二元组中也很明显。仅使用字母组合，就不会分别与d或i匹配。另一方面，使用双字母组，甚至可以校正非常复杂的分割错误，如图3所示。

图3.基于图像的前后校正的示例。

包含二元组也引入了一个新问题。

双字和单字可能会在字符是否应该更正方面存在分歧。通过允许二元组推翻unigram聚类类型，可以解决此类分歧。对于每个Master字母组合，如果足够数量的样本包含在“拒绝”二元组中，则该字母组合也将标记为“拒绝”。相反，如果“拒绝”单字组中的大多数样本都在“主”二元组中，则该“单字组”将重新键入为主。

C.假设验证当基于语言的校正路径生成一个

假设，在接受更改之前，需要通过图像模型对其进行验证。该更改首先分解为一组原子操作，例如符号插入/删除，大小写折叠，空格插入/删除或替换。例如，从“ fmanCial”到“ financial”的更正将导致替换{m→in}和案例折叠{C→c}。如果样本m来自一个小簇，或者它到簇中最近的距离小于阈值，则替换更改是可以接受的。为了验证案例折叠的变化，我们使用相对于其相邻符号的包围盒轮廓以及C和c之间的簇距离。为每种修改类型定义了相似的规则。如果没有任何组成操作被拒绝，则接受更改。找到完美的解决方案以协调这两个模型之间的置信度得分不是目标，而只是拒绝基于图像特征的不太可能的更改。

IV。基于语言的纠正

语言的组织和功能

校正路径类似于图像路径。将归一化的单词标记视为一个群集，将在上下文中一起评估所有相同标记的出现，以尝试找到可能导致更高可能性的替代方法。如果找到替代方法，则将针对每个实例分别针对语言和图像模型进行评估，以确定是否应接受更正。

A.自适应语言模型我们的基本语言模型由单词n-gram和

在大型Web语料库上训练的词汇外术语回退到字符n-gram。由于一本书通常包含成千上万个单词，并且分布非常不同，尤其是在诸如专有名词或主题指示符之类的内容关键字上，因此使用自适应缓存模型进行扩充非常重要。

考虑到目标是纠正较低的先验

在基本模型中内容关键字的概率方面，我们使用了一个简单的词频表作为缓存模型。模型被初始化为空，并且通过三个条件之一将单词添加到模型。第一个来源是OCR引擎。如果单词被引擎标记为词典单词或具有很高的置信度，则该单词将被接受到模型中。第二个来源是基础语言模型。将添加由基本单词模型验证的任何单词（无校正假设）。为了解决基本模型中没有的内容关键字的情况，我们还包括高频OOV（语音外）词，这些词在语言模型中没有可行的选择，并且具有很高的置信度。我们计算这些单词相对于书中单词总数的频率。

有几种方法可以将基本缓存与缓存结合起来

模型，例如加权平均或最大熵。我们发现简单地采用最大值比其他更复杂的加权方案更好。如果令牌在高速缓存模型中具有更高优先级的条目，则这将覆盖基本模型概率P（w）= max {Pcache（w），Pbase（w）}。尽管这不能正确地调整条件概率

如果评估包含w的较大上下文的可能性，则仅调整先验似乎是非常有效的。

B.假设的产生为了识别潜在的错误，我们想隔离

根据我们的模型，可能性较低的单词或句段。此外，我们还希望生成比当前标签更有可能的校正候选。为了有效地做到这一点，我们共同考虑了每个OOV词w在其各自上下文中的所有出现。更准确地说，假设Ci（w）是一个以w的第i个实例为中心的n = 7个单词的窗口，考虑到识别输出w及其上下文Ci（w），我们希望找到最可能的单词标签q，

齐* = argmaxq P（q | w，Ci（w））

~argmaxq logP（w | q，Ci（w））+ logP（q，Ci（w））

其中，P（q，Ci（w））是给定上下文的语言模型似然性，P（w | q）是体现混淆概率的噪声通道[4]，我们假设其独立于相邻单词。使用维特比搜索，我们可以找到最可能的候选气

*在给定的上下文中。如果齐 * = w，表示没有比当前标签更好的答案了。对每个上下文重复该过程将产生候选q1

*.. qN *.

不幸的是，由于我们的缓存模型未与在基本模型中，似然性P（q，Ci（w））的计算或维特比搜索都不会正确地说明该调整。因此，该模型将不公平地惩罚比基础模型具有更高先验性的单词，并且在搜索过程中不太可能找到该单词作为首选单词。我们使用对数线性公式对此进行近似，其中可能性基本上是分别计算然后求和的。

齐* = argmaxq {α（logPbase（w | q）+ logPbase（q，Ci（w））+

β（logPcache（w | q）+ logPcache（q））}

其中Pcache（q）是上述文档频率，而Pcache（w | q）是静态OCR混淆表，其中增加了经过验证的文档更正。没有尝试学习设置为1的α和β。

一个简单的策略是用qi代替w *每个实例独立。但是，这种方法经常会因缺少简短或模棱两可的情况下缺乏足够证据而错过校正。因此，我们对所有上下文的似然比求和，以得出最可能的答案，并对所有实例应用相同的校正。

这在图4中示出。我们评估令牌这个词“ thinx”在四个不同的上下文中。假设维特比搜索产生两次“思考”，一次产生“感谢”，这是更好的选择，而在另一次中没有提出建议。每个候选者与基本假设的似然比在所有情况下求和，然后重新映射为置信度以产生最佳选择“思考”。为了避免出现不同情况的正确答案确实不同的情况，如最后一个示例“非常感谢”，我们针对P（w，Ci（w）检查似然性P（q *，Ci（w））））），并且只有在可能性提高的情况下，才产生校正假设。

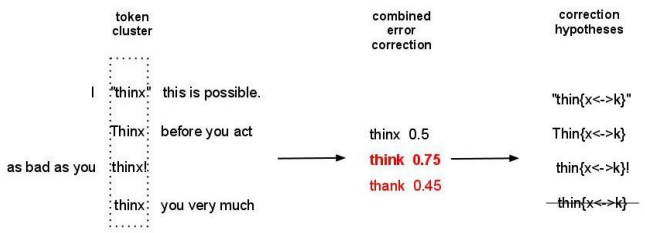

图4.组合纠错。

C.案例推理一个值得讨论的实际问题是令牌规范化。

尽管可以将所有案例变体，标点和连字吸收到一个模型中，但是通常在语言模型中将这些变体标准化。令牌将经过预处理后进入模型，并且返回时会更正校正。不幸的是，Tesseract经常会出现很多套管错误，这通常是由于在类似形状对（例如c / C，w / W）上的错误匹配造成的。使用诸如TrueCase [6]中所述的信息，例如句子边界，单词上下文，大写规则，可以推断给定单词最可能的表面形式。尽管这为仅使用文本信息的单词大小写提供了很好的先验，但它不能解决源自图像的强烈信号。例如，在相同的上下文中，诸如“ DVD”之类的单词可能大写或不大写。仅使用根据文本信息计算出的概率将在所有情况下强制使用相同的表面形式，并且不可避免地会产生许多错误。

最强的信号来自图像，即通过OCR引擎提供的初始答案间接观察到。我们基于原始OCR输出到每个曲面形式之间的编辑距离成本添加了一个功能。另一个功能是基于将表面形式的预期单词形状与实际边界框轮廓进行比较。由于书头经常被大写或全部用大写，因此估计书头块具有不同的先验分布。

使用与生成假设相同的方法考虑到上下文，我们学会了一个模型，可以使用这些信号来推断最可能的大小写。纠正拼写错误后，我们在每个实例上应用大小写推断作为非规范化过程的一部分。

五，评估与结果

该系统在扫描书籍上进行了评估，与质量要比Google1000数据集中的质量高[9]。在开发过程中使用了大约200万个单词（1100万个字符）的Smoke集进行错误分析和系统调整。最终测试是在更大的600万个单词的测试集上进行的。在网络语料库上训练了基本语言模型，并从拼写错误中学习了噪音通道。对于案例推理模型，我们从Google Ngram Viewer数据中收集了统计数据[12]，并在烟具上调整了几何特征。

在继续实验结果之前，我们应该解释评估过程和指标。由于很难手动收集和验证这种大小的数据集，因此我们依靠半自动方法，将扫描的书籍与可用的PDF来源对齐。造成此事实数据不完善的因素有很多，包括重复或重复。

扫描过程中出现乱序的页面，输出序列中的块顺序不同，PDF和扫描版本之间的版本不匹配等。因此，我们开发了一种基于字符串对齐的自动评估方法。因此，由于对齐段的差异，实验之间存在差异。

定义了大量指标以帮助衡量各种

结果的各个方面。这里总结了三个最重要的指标。 ch.subst测量文本的对齐部分中的字符替换错误率。此措施区分大小写和标点符号。 wd.err比率基于去除了标点符号的大小写折叠的标记，不包括停用词，但包括单词替换，插入和删除。从索引和搜索的角度来看，该措施很有意义，但更容易受到对齐差异的影响。为了减轻由于对齐导致的影响，flwd.drop速率将基本事实视为一袋单词，并计算OCR输出中缺少的基本事实单词的百分比。

结果总结在表1中。使用Tesseract

作为基线的输出，这些列显示了通过建议的方法LIM获得的每个指标的相对变化百分比。作为参考，还给出了仅使用基于图像的校正（IM）和仅基于语言的校正（LM）产生的改进。

在烟具上，ch.subst，wd.err和flwd.drop速率

分别下降了36.88％，22.58％和24.39％。 IM和LM的结果表明，两个校正分支都做出了重大贡献。字符替换率的其他改进主要归因于修复了大小写错误和同一单词中的多个错误。在测试中，字错误率降低了18％。 IM和LM在此较大集合上产生的减少较少。然而，ch.subst的减少幅度要小得多，仅为13％。相对于12％的字错误减少，从ch.subst到LM的4.85％的减少要小得多，这表明案例推断模型在该集合上的推广效果很差。

表I.基准TESSERACT上的系统性能改进。

数据集模型Δch.subst%Δwd.err%Δflwd.drop%				
烟雾IM -7.92 -7.5 -7.47				
	LM -30.47 -17.74 -19.23			
	林-36.88 -22.58 -24.39			
测试IM -6.77 -3.19 -3.14				
	LM -4.85 -10.56 -12.04			
	林-13.58 -16.11 -18.16			

更详细的分析表明，提出的解决方案

除有关统计物理学和蛋白质折叠的书外，几乎每本书都进行了改进，其中案例推理模型过分用力地纠正了包含混入大小写的单词的不良识别方程，最终导致了更多的替换错误。但是，总的flwd.drop率仍然有所提高，这意味着拼写错误的单词已得到纠正。另一个失败案例发生时，一个重音外国名称被一致认为是一个更通用的英文名称。这表明自适应模型需要改进，图像模型验证需要加强。

我们以占总CPU时间的百分比来衡量运行时间

由Tesseract基于同一数据集。由于自适应图像模型需要对所有unigram和bigram段进行聚类和分类，因此运行时间会因图像质量和页面内容而异，在Smoke上占55％，在Test上占84％。由于语言校正路径采用了netword分布式服务，因此更难测量语言校正路径的运行时间。我们估计，除OCR之外，总的语言CPU开销为10-15％。

VI。结论

我们提出了一个包含两个修正的系统

基于特定于文档的图像和语言模型的路径。每个自适应模型都利用字体和词汇表中的冗余来检测不一致之处，但是利用另一个模型的正交性来验证校正假设。在大型测试集上，该系统能够将Tesseract的单词错误率降低25％。

总体而言，我们对结果的考虑感到非常鼓舞

系统中采取的众多简化步骤。我们认为，解决其中一些问题可以取得重大进展。例如，基础语言校正模型是使用查询拼写错误作为噪音通道在网络语料库上训练的；显然，应该使用书籍语料库和OCR混淆来对其进行重新培训。同样，将文档语言模型更好地集成到Viterbi搜索过程中应该会产生更好的更正。此外，最小错误训练可以应用于微调系统参数，例如图像模型验证阈值和对数线性系数。除大小写推断模型外，已在其他拉丁语系语言上尝试了相同的策略，但取得了类似的成功。推广使用非字母或非基于单词的语言的方法仍然是未来的工作。

参考资料

[1] T.K. Ho，G. Nagy，“不进行形状训练的OCR”，ICPR，第27-30页，2000年。

[2] A. Kae，G. Huang，C. Doersch，E.Learned-Miller，“通过高精度的特定于文档的建模改进最先进的OCR”，CVPR，第1935-1942年，2010年。

[3] A. Kae，E。Learned-Miller，“动态学习：困难OCR问题的无字体方法”，ICDAR 2009。

[4] O. Kolak，P。Resnik，“使用噪声通道模型的OCR纠错”，人类语言技术大会，2002年。

[5] C.J. Leggetter，P.C. Woodland，“用于连续密度隐藏马尔可夫模型的说话人适应的最大似然线性回归”，《计算机语音与语言》，第9卷第2期，第171-185页，1995年。

[6] L.V.Lita，A。Ittycheriah，S。Roukos，N。Kambhatla，“ tRuEcasIng”，ACL，2003年。

[7] P. Sarkar，G。Nagy，“同质模式的样式一致分类”，IEEE Tans。于PAMI，2005年1月27（1）。

[8] R. Smith，“ Tesseract OCR引擎概述”，ICDAR，2007年。

[9] L. Vincent，“ Google图书搜索：大规模的文档理解”，ICDAR，2007年。

[10] S. Veeramachaneni，G。Nagy，“多源自适应分类器
OCR”，IJDAR，第6卷，第154-166页，2003年。

[11] P Xiu，H Baird，“迈向全书认可”，DAS，第629-636页，
2008。

[12] http://ngrams.googlelabs.com/datasets