福昕PDF编辑器

• 永久　• 轻巧　• 自由

升级会员　　批量购买

永久使用

无限制使用次数

极速轻巧

超低资源占用，告别卡顿慢

自由编辑

享受Word一样的编辑自由

扫一扫，关注公众号

从图像中检测和识别表格，北航&微软提出新型数据集TableBank

机器之心 3/21

选自 arxiv

作者：Minghao Li 等

机器之心编译

机器之心编辑部

该研究中，来自北航和微软亚研的研究者联合创建了一个基于图像的表格检测和识别新型数据集 TableBank，该数据集是通过对网上的 Word 和 Latex 文档进行弱监督而建立的。该数据集包含 417K 个高质量标注表格，通过此数据集作者利用深度神经网络 SOTA 模型建立了数个强大的基线，从而助力更多研究将深度学习方法应用到表格检测与识别任务中。目前 TableBank 已开源。

TableBank 开源地址：https://github.com/doc-analysis/TableBank

表格通常以结构化的方式展示基本信息，因而表格检测和识别是诸多文件分析应用中的一项重要任务。如图 1 所示，由于表格的布局和格式不同，其检测和识别是个难题。常规表格分析技术通常以文件的布局分析为基础。但这些技术中的大多数都无法泛化，究其原因，它们依赖于手工构建的特征，而后者对布局变化不具备稳健性。最近，计算机视觉领域深度学习的快速发展极大地推动了数据驱动且基于图像的表格分析方法。基于图像的表格分析的优势体现在其对文件类型的稳健性，并对文件是页面扫描图像还是原始数字文件格式不做任何假设。因此，大型端到端深度学习模型能够取得更好的效果。

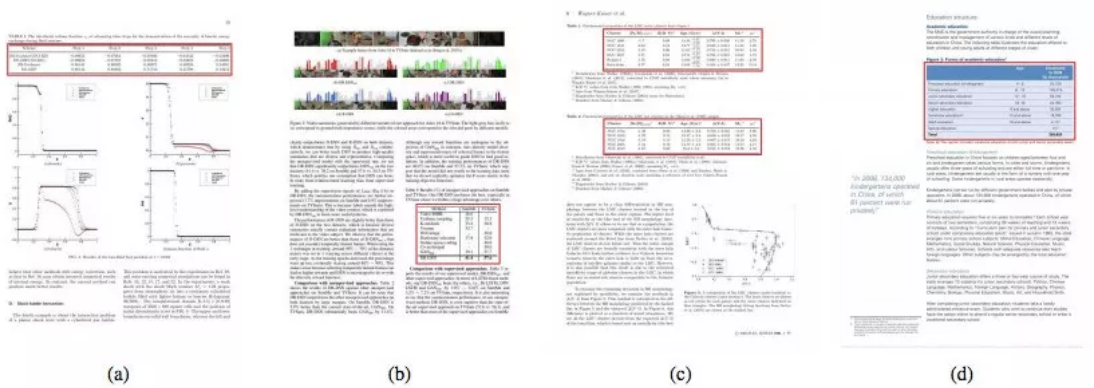

图 1：不同布局和格式的表格电子文件。

现有的基于深度学习的表格分析模型通常对使用数千个人工标注训练实例获得的预训练目标检测模型进行微调，但它依然难以在现实世界应用程序中扩展。例如，我们发现，在类似图 1a、1b 和 1c 中的数据上训练出的模型在图 1d 中表现不佳，其原因在于表格布局和颜色大不相同。因此，扩大训练数据是使用深度学习构建开放域表格分析模型的唯一途径。深度学习模型比传统模型复杂得多，现在的很多标准深度学习模型拥有数亿自由参数，且需要更多标注训练数据。在实践中，人工标注大型训练数据成本高昂且缺乏灵活性，这是实际部署深度学习模型的关键瓶颈。众所周知，ImageNet 和 COCO 是两个流行的图像分类和目标检测数据集，两者均以众包的方式构建，但花费高昂且耗日持久，需要数月甚至数年时间来构建大型基准数据集。幸运的是，网络上存在大量数字文件，如 Word 和 Latex 源文件。对这些在线文件进行一些表格标注方面的弱监督则是有益的。

为解决对标准开放域表格基准数据集的需求，该研究提出一种新颖的弱监督方法，可自动创建 TableBank 数据集，TableBank 要比现有的表格分析人工标注数据集大几个量级。与传统弱监督训练集不同，该研究提出的弱监督方法可以同时获得大规模和高质量的训练数据。现在，网络上有大量电子文档，如 Word（.docx）和 Latex（.tex）文件。这些在线文档的源代码中包含表格的 mark-up tag。直观地讲，借助每个文档中的标记语言，研究者可以通过添加边框来操控这些源代码。就 Word 文档而言，内部 Office XML 代码可以在标注每一表格边界的地方进行修改。就 Latex 文档而言，tex 代码同样可以在标注表格边界的地方进行修改。这种方式可以为多个不同域创建高质量的标注数据，如商业文件、官方名录和科研论文等，这些数据对大规模表格分析任务大有裨益。

TableBank 数据集共包含 417,234 个高质量标注表格以及各域中对应的的原始文档。为验证 TableBank 的效果，研究者使用当前最优的端到端深度神经网络模型构建了多个强大的基线。表格检测模型基于不同设置下的 Faster R-CNN 架构（Ren 等人，2015 年），表结构识别模型基于图像-文本（image-to-text）的编码器-解码器框架。实验结果表明，布局和格式变化对表格分析任务的准确率影响很大。此外，在某一特定域训练的模型在另一域中表现不佳。这表明，在 TableBank 数据集上建模和学习还有很大的进步空间。

数据收集

大致上，研究者构建 TableBank 数据集时使用了两种不同的文件类型：Word 文档和 Latex 文档。这两种文件类型的源代码中都包含 mark-up tag。这部分分三步详细介绍了数据收集过程：文档获取、创建表格检测数据集、创建表结构识别数据集。

文档获取

研究者从网上抓取 Word 文档。这些文档都是 .docx 格式，因此研究者可以通过编辑内部 Office XML 代码来添加边框。研究者并未过滤文档语言，因此这些文档包含英语、中文、日语、阿拉伯语和其他语言。这使得该数据集在实际应用中更多样化、更稳健。

Latex 文档与 Word 文档不同，因为前者需要其他资源来编译成 PDF 文档。因此，研究者不能从网上抓取 tex 文档，而是利用最大预印本数据库 arXiv.org 中的文档以及相应的源代码。借助 arXiv bulk data access，研究者下载了 2014 年至 2018 年论文的 Latex 源代码。

表格检测

直观地讲，借助每个文档中的标记语言，研究者可以通过添加边框来操控源代码。处理流程如图 2 所示。就 Word 文档而言，研究者通过编辑每个文档中的内部 Office XML 代码来添加表格边框。每个 .docx 格式文件有一个压缩包，解压后的文件夹中有一个 document.xml 文件。在 XML 文件中，该代码片段介于标记 <w:tbl> 和 </w:tbl> 之间，通常表示 Word 文件中的表格，如图 3 所示。研究者修改 XML 文件中的代码片段，使表格边框可更改为与文档其他部分不同的颜色。如图 3 所示，研究者在 PDF 文档中添加了一个绿色边框，该表格得到完美识别。最后，研究者从 Word 文档中获得了 PDF 页面。

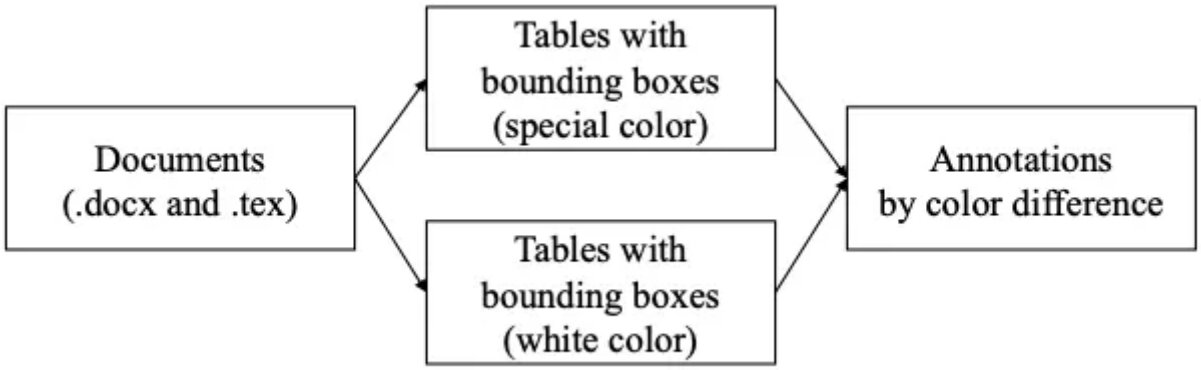

图 2：数据处理流程。

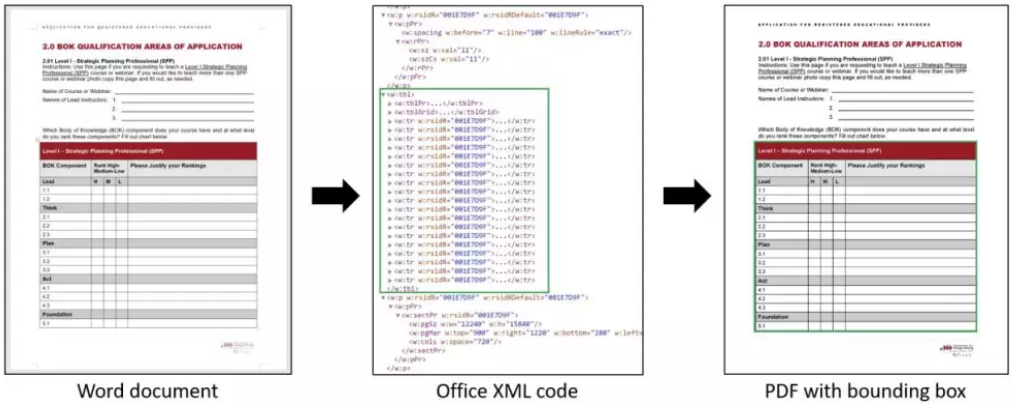

图 3：通过 Office XML 代码中的 <w:tbl> 和 </w:tbl> 标记来识别和标注表格。

表结构识别

表结构识别旨在确定表格的行列布局结构，尤其适用于扫描图像等非数字化文档格式的表格。现有表结构识别模型通常用于识别布局信息和单元格的文本内容，而文本内容识别并非这一工作的重心。所以，研究者将任务定义为：给定一个图像格式的表格，生成表示表格行列布局和单元格类型的 HTML 标签序列。通过这种方式，研究者可以从 Word 和 Latex 文档的源代码中自动构建表表结构识别数据集。就 Word 文档而言，研究者只需将原始 XML 信息从文档格式转换成 HTML 标签序列即可。而对于 Latex 文档，研究者首先使用 LaTeXML toolkit 从 Latex 中生成 XML，然后将其转换为 HTML 格式。如图 4 中的简单示例，研究者使用 <cell_y> 表示含有文本的单元格，<cell_n> 表示没有文本的单元格。在过滤噪声后，研究者基于 Word 和 Latex 文档创建了 145,463 个训练实例。

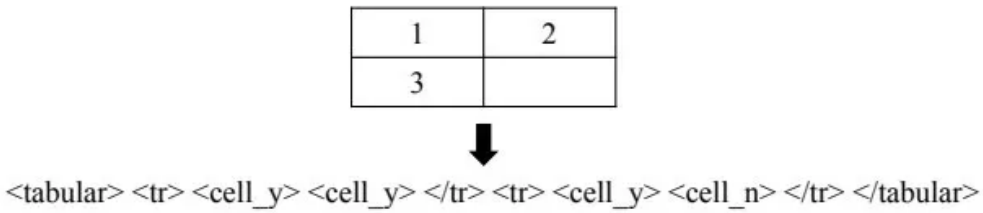

图 4：表格转 HTML 示例，其中 <cell_y> 表示含有文本的单元格，<cell_n> 表示没有文本的单元格。

基线

表格检测

该研究使用 Faster R-CNN 作为表格检测基线模型，其架构如下图所示：

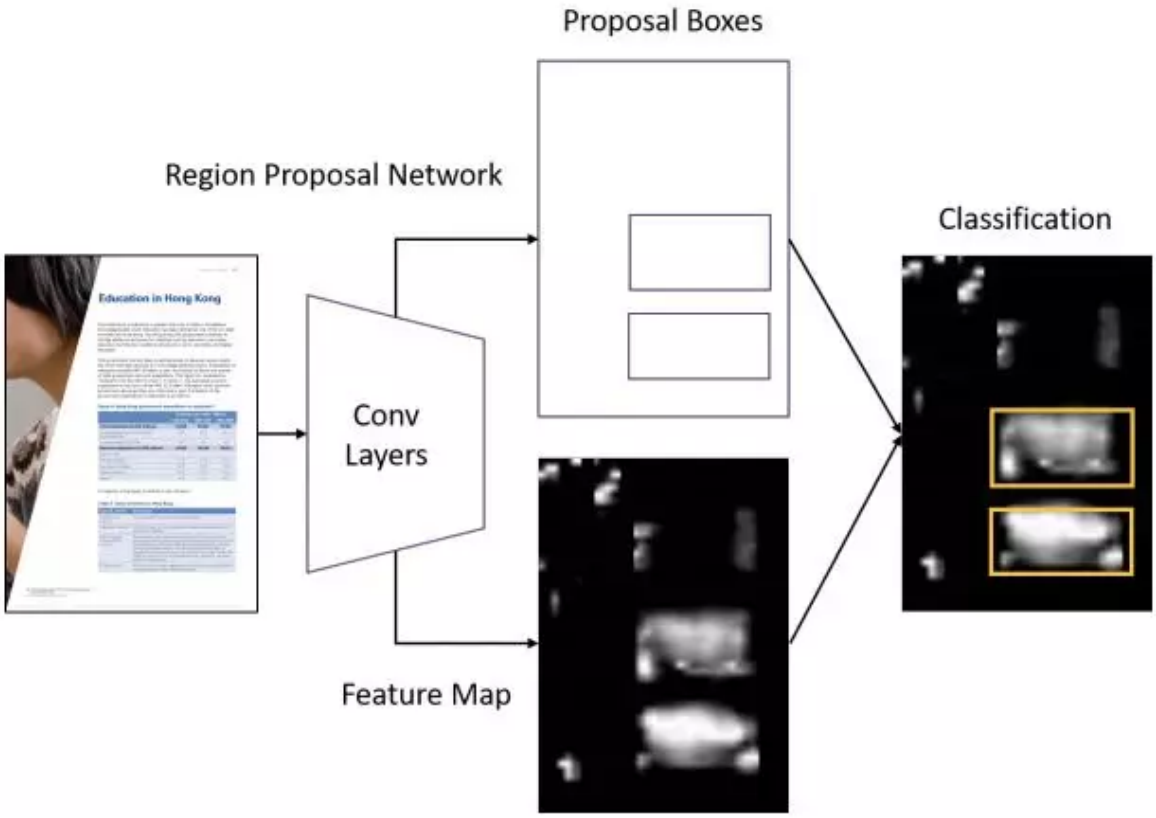

图 5：用于表格检测的 Faster R-CNN 模型。

表结构识别

该研究使用图像-文本模型作为表结构识别的基线模型，其整体架构如下图所示：

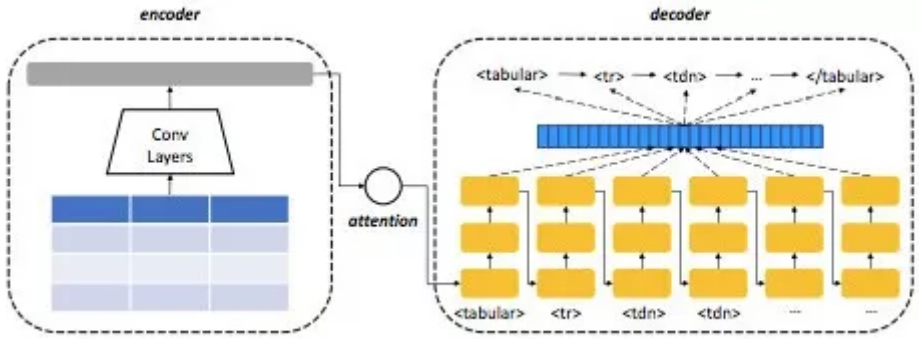

图 6：用于表结构识别的图像-文本模型。

实验

Task	Word	Latex	Word+Latex
Table detection	163,417	253,817	417,234
Table structure recognition	56,866	88,597	145,463

表 1：TableBank 数据集的统计数据。

Models	Word			Latex			Word+Latex		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
ResNeXt-101 (Word)	0.9496	0.8388	0.8908	0.9902	0.5948	0.7432	0.9594	0.7607	0.8486
ResNeXt-152 (Word)	0.9530	0.8829	**0.9166**	0.9808	0.6890	0.8094	0.9603	0.8209	0.8851
ResNeXt-101 (Latex)	0.8288	0.9395	0.8807	0.9854	0.9760	0.9807	0.8744	0.9512	0.9112
ResNeXt-152 (Latex)	0.8259	0.9562	0.8863	0.9867	0.9754	**0.9810**	0.8720	0.9624	0.9149
ResNeXt-101 (Word+Latex)	0.9557	0.8403	0.8943	0.9886	0.9694	0.9789	0.9670	0.8817	0.9224
ResNeXt-152 (Word+Latex)	0.9540	0.8639	0.9067	0.9885	0.9732	0.9808	0.9657	0.8989	**0.9311**

表 2：使用 ResNeXt-{101,152} 作为骨干网络对 Word 和 Latex 数据集的评估结果。

Models	Word	Latex	Word+Latex
Image-to-Text (Word)	**0.7507**	0.6733	0.7138
Image-to-Text (Latex)	0.4048	**0.7653**	0.5818
Image-to-Text (Word+Latex)	0.7121	0.7647	**0.7382**

表 3：图像-文本模型在 Word 和 Latex 数据集上的评估结果（BLEU）。

Length	0-20	21-40	41-60	61-80	>80	All
#Total	32	293	252	145	278	1,000
#Exact match	15	169	102	28	24	338
Ratio	0.469	0.577	0.405	0.193	0.086	0.338

表 4：生成 HTML 标注序列和真值序列之间的精确匹配（exact match）数量。

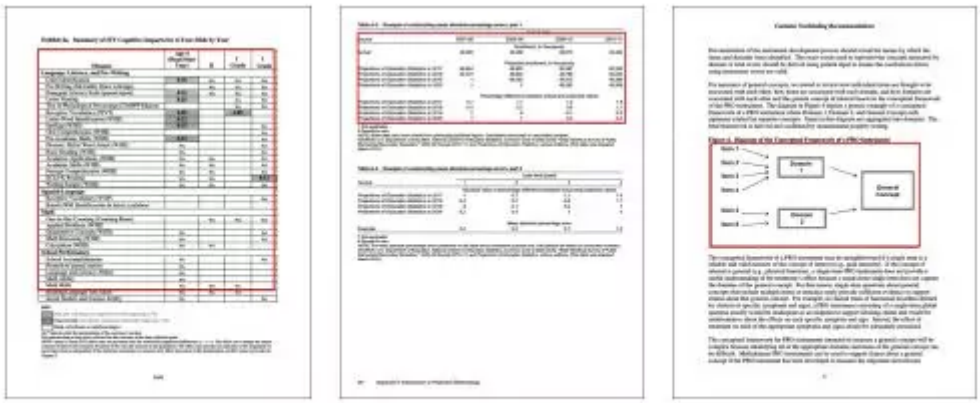

图 7：使用 a）partial-detection、b）un-detection 和 c）mis-detection 进行表格检测的示例。

论文：TableBank: Table Benchmark for Image-based Table Detection and Recognition