a. 若其中一个字符的 x1 与下一个字符的 x0 之差大于 x_tolerance，则添加空格。

b. 若其中一个字符的 doctop 与下一个字符的 doctop 之差大于 y_tolerance，则添加换行符。

Ⅳ. extract_tables(table_settings) 方法：从页面中提取表格数据

①获取 Table：.find_tables(table_settings={})

返回 Table 对象的列表。

Table 对象提供对.cells，.rows 和.bbox 属性以及.extract(x_tolerance = 3, y_tolerance = 3)方法的访问。

②提取表格文本：.extract_tables(table_settings={})

返回从页面上找到的所有表中提取的文本，并以结构 table -> row -> cell 的形式表示为列表的列表。

即：每一个表格为一个列表(table)，每个列表的每一行数据为一个列表（row),每行数据的列表中的每个元 素为一个单元格的数据（cell）

③提取最大表格的文本：.extract_table(table_settings={})

返回从页面上最大的表中提取的文本，以列表列表的形式显示，结构为 row -> cell。

（如果多个表具有相同的大小——以单元格的数量来衡量——此方法将返回最接近页面顶部的表

④获取表格信息：.debug_tablefinder(table_settings={})

返回 TableFinder 类的实例，可以访问.edges，.intersections，.cells 和.tables 属性

pdfplumber 是一个开源的 Python 库，专门用于读取、解析和操作 PDF 文件。它允许用户以编程方式提取 PDF 文档中的文本、图像、表格以及其他各种元素，并且能够保留原始布局信息。

通过 pdfplumber，您可以实现以下功能：

　　文本抽取：逐字符或逐段落获取 PDF 中的文本内容，包括字体大小、颜色等样式信息。

　　图像提取：从 PDF 中导出图片到本地文件。

　　表格识别与处理：将 PDF 中的表格结构转换为数据结构（如 pandas DataFrame），方便进一步的数据分析和处理。

　　精确测量：获取页面上任意元素的位置、尺寸以及边界框信息。

　　图形对象支持：对于线条、矩形等基本图形也能进行提取和测量。

使用 pdfplumber 时，您只需打开 PDF 文件，然后遍历每一页并调用相关方法来获取所需内容。由于其对复杂版面的良好适应性，pdfplumber 在处理非结构化或半结构化的 PDF 文档时表现优秀，尤其在数据抓取、自动化报告处理等领域有着广泛的应用。