PDF 解析：从视觉到结构的重建之旅

引言

在你辛辛苦苦编辑完一个报告、一份简历、一份论文后，是否习惯将其保存为 PDF 文件？这样做的好处显而易见：一旦生成，文档的格式和视觉效果就固定不变，**无论在哪种应用程序、硬件、操作系统或打印设备上，均能一致呈现**。

然而，这种优势也是一把双刃剑。当你想从 pdf 文件中再还原之前的文档或页面时，则相当困难，为什么会这样呢？那就需要我们来探究一下 PDF 文档的原理。

PDF 的原理与解析挑战

在建设 RAG（Retrieval-Augmented Generation）能力时，解析 PDF 文件显得尤为重要。RAG 需要从大量以 pdf 为主的文档中提取和理解信息，而 PDF 文件的视觉数据限制了直接获取内容的能力。因此，解析 PDF 以重建其逻辑结构，是实现高效信息检索和生成的关键一步。

PDF 文件最初由 Adobe 公司发明，旨在解决跨平台文档共享的问题。其设计目的是确保文档在不同设备和操作系统上都能**保持一致的外观和格式**。

为了确保将文档的视觉呈现固定化，PDF 仅记录页面上字符、线条的位置、颜色和大小等视觉信息，却牺牲了文档的结构信息。结构信息指的是文档内容的组织方式和语义信息，例如章节、段落、标题、列表等。这些结构信息对于 PDF 文档的进一步分析和再利用至关重要。

我们来看下图中的文档：

第 36 卷 第 9 期
2022 年 9 月
中文信息学报
JOURNAL OF CHINESE INFORMATION PROCESSING
Vol. 36, No. 9
Sep., 2022

文章编号：1003-0077(2022)09-0001-18

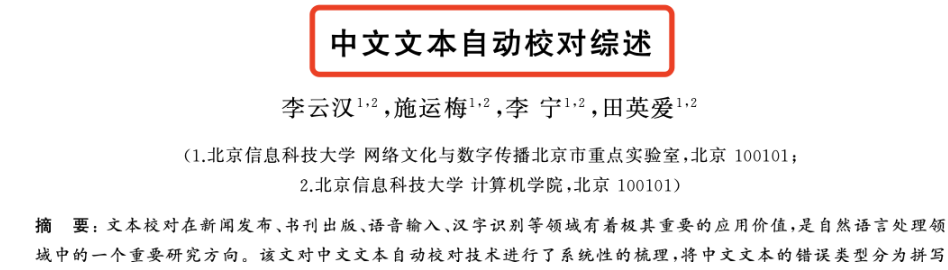


图 1：中文论文标题

对于图中的论文标题，pdf 文件存储的信息可理解为：

> 在(x:219, y:122)位置
> 用宋体、大小为 18 号、黑色的、粗体字
> 绘制一个内容为"中文文本自动校对综述"的文字框

所以与其说 PDF 是一种数据格式，不如说它是印刷指令的集合更为准确。通过 PDF 文件存储的内容，计算机是无法直接得知“这个文字框是否为标题”，“文字属于哪个表格”，“这个段落是否跨栏或者跨页了”等结构信息的。因此，如果想让计算机“真正弄懂”这种"仅关注外在"的文档，以用于后续的内容分析和再利用，必须完成的一个任务就是：

从多页且复杂排版、仅提供视觉数据的 PDF 文件中，重建出一个完整而丰富的包含逻辑结构的文章内容。

相关工作

在 PDF 解析领域，业界的方法一般归为以下几类

1、基于规则的解析方法：

优点：解析准确率较高，尤其在特定领域和业务场景中

缺点：在布局分布丰富的场景中通用性欠佳，难以保证用预定义的规则涵盖所有文件类型和布局。

2、基于视觉模型的解析方法：主要以 unstructured、surya、chunkr 为主

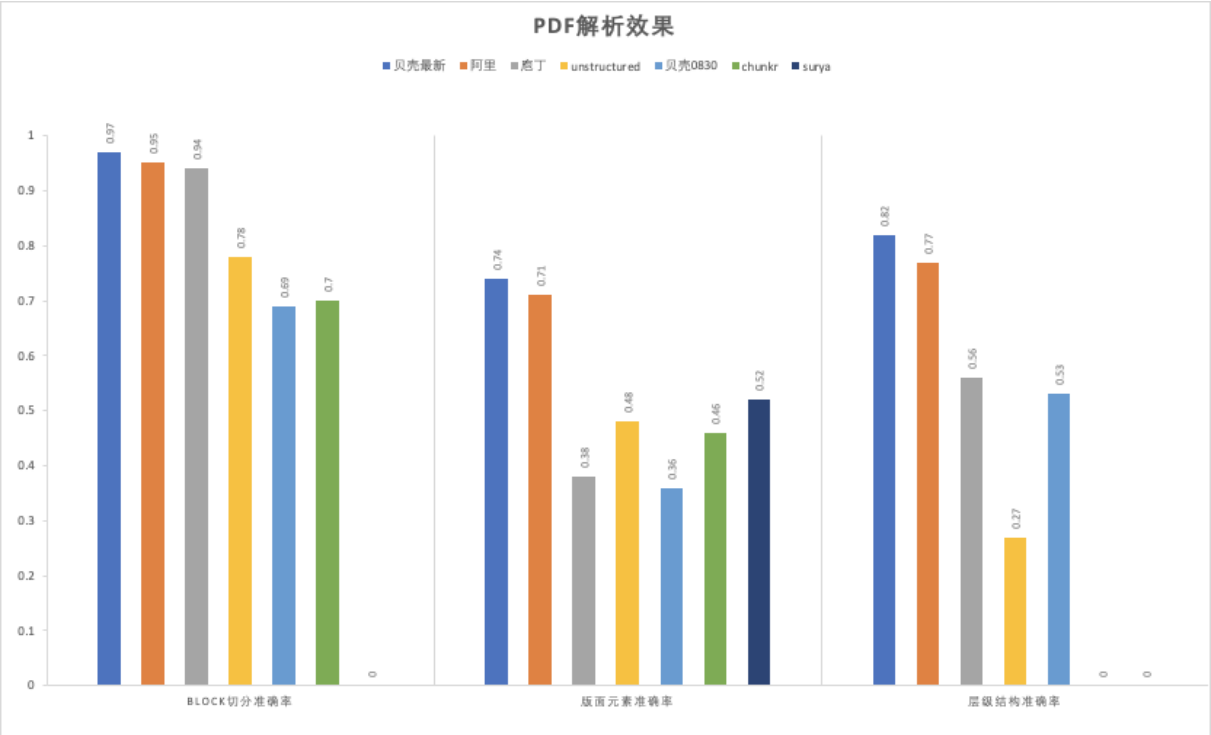
优点：通用性更强，

缺点：该类方法一般只能对文档内容做切分和版面元素的识别，但解析结果中没有各个块间的

逻辑关系。另外因其基于 OCR 模型，则无法避免字符的识别误差；

3、以上两种方式的结合，主要以主要以阿里、庖丁为代表；

我们对业界常见的解析引擎和贝壳自研的解析引擎，进行了效果评测，结果如下：



评测时间 2024 年 11 月 25 日

图中可以看出，在 block 切分准确率等三个指标中，贝壳自研的 pdf 解析引擎都处于领先位置。那么，评测方法是否足够客观准确呢？让我们一起来看一下整个评测体系的建立过程。

PDF 解析评测

构建评测集

1、**了解文档布局**：首先，我们调研了 DocLayNet 文档集，以了解常见文档布局的丰富程度和分布情况。该数据集包含了 8 万个页面布局，为评测集的构建提供了广泛的参考。

2、**确定评测广度**：明确需要覆盖的解析点边界，确保评测集能够涵盖多种文档结构和内容类型。这包括图像、表格、段落、公式、代码块、目录等 **30+类解析评测点**，具体如下：

- 图像：图片、图注、图题、线框图
- 表格：表格、表注、表题、**非闭合表、单元合并、表格跨页**
- 段落：**段落跨页、多列**
- 标题：标题

- 列表：有序列表、无序列表
- 公式：段落公式、独立公式
- 代码块：代码块
- 目录：目录
- 无用信息：封面、序言、附录、后记、参考文献
- 页眉页脚：单字、多字段文字式、图文混合式、奇偶页变化式、含页码式页眉页脚

3、**业务文档采样**：从实际业务文档中进行采样，选择能够涵盖已确定解析类别的文档。这一步确保评测集具有广泛的案例覆盖度、现实应用的代表性。

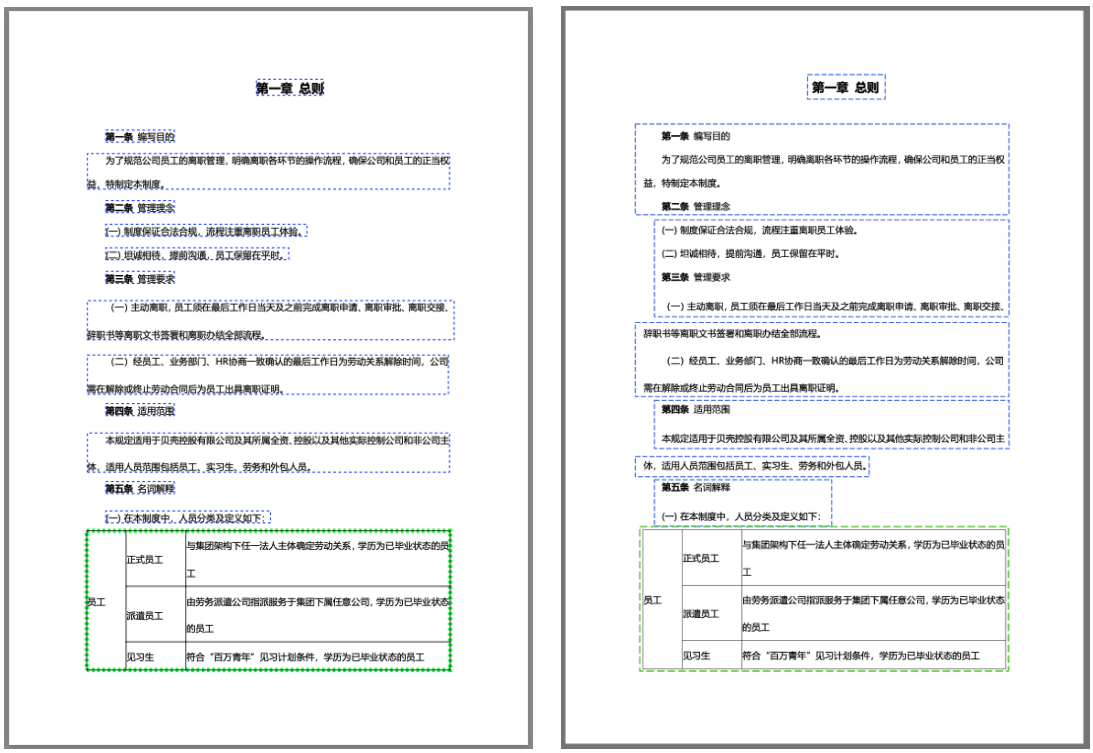
4、**解析块构建**：在采样的文档中，人工识别并标注了 **453 个解析块的内容、块间的层级关系**。这些解析块涵盖了解析评测点中的各种版面元素，确保评测集的全面性。

设计评测指标

在评测集构建完成后，我们需要明确评估解析引擎效果的角度，并定量输出解析效果指标。

1、片段切分准确率

首先，最直观的评估角度是判断文章片段的切分是否合理。逻辑结构一致的内容应被归入同一片段。



较好分块 vs 较差分块

我们将人工标注的标准分块结果作为 Groud Truth，与解析引擎的分块结果做对比，用编辑距离满足阈值的片段的占比作为切分合理性的衡量。

故该指标定义如下：

$$\text{片段切分准确率} = \frac{1}{n}\sum_{n=1}^{\infty}\begin{cases}1-\frac{\text{编辑距离}}{\text{节点字符长度}} & \text{建立 } 1v1 \text{ 映射关系}\\ 0 & \text{其他}\end{cases}$$

n 为总 label 节点数

2、 版面元素正确率

其次，准确输出某个片段在文章中属于何种版面元素也很重要。例如，判断片段是标题、表格、注释、普通段落，还是非正文内容如页眉页脚、目录等，是衡量解析引擎文档理解深度的重要标准。

下图展示了我们对理想解析引擎的期望：不仅要实现正确的片段切分，还需准确判断片段的版面元素类型：

Title

这是一个自制的文件

Title

第一部分 文字图片

Text

这是文字这是文字，这是文字这是文字。这是文字这是文字，这是文字这是文字。这是文字这是文字，这是文字这是文字。这是文字这是文字，这是文字这是文字。这是文字这是文字，这是文字这是文字。这是文字这是文字，这是文字这是文字。

Figure

图 1-1

Text

这是文字这是文字，这是文字这是文字。这是文字这是文字，这是文字这是文字。这是文字这是文字，这是文字这是文字。这是文字这是文字，这是文字这是文字。这是文

那么该指标定义为：

$$\text{版面识别准确率} = \frac{1}{n}\sum_{n=1}^{\infty}\frac{1}{m}$$

n 为在解析结果中能找到映射的 label 节点数

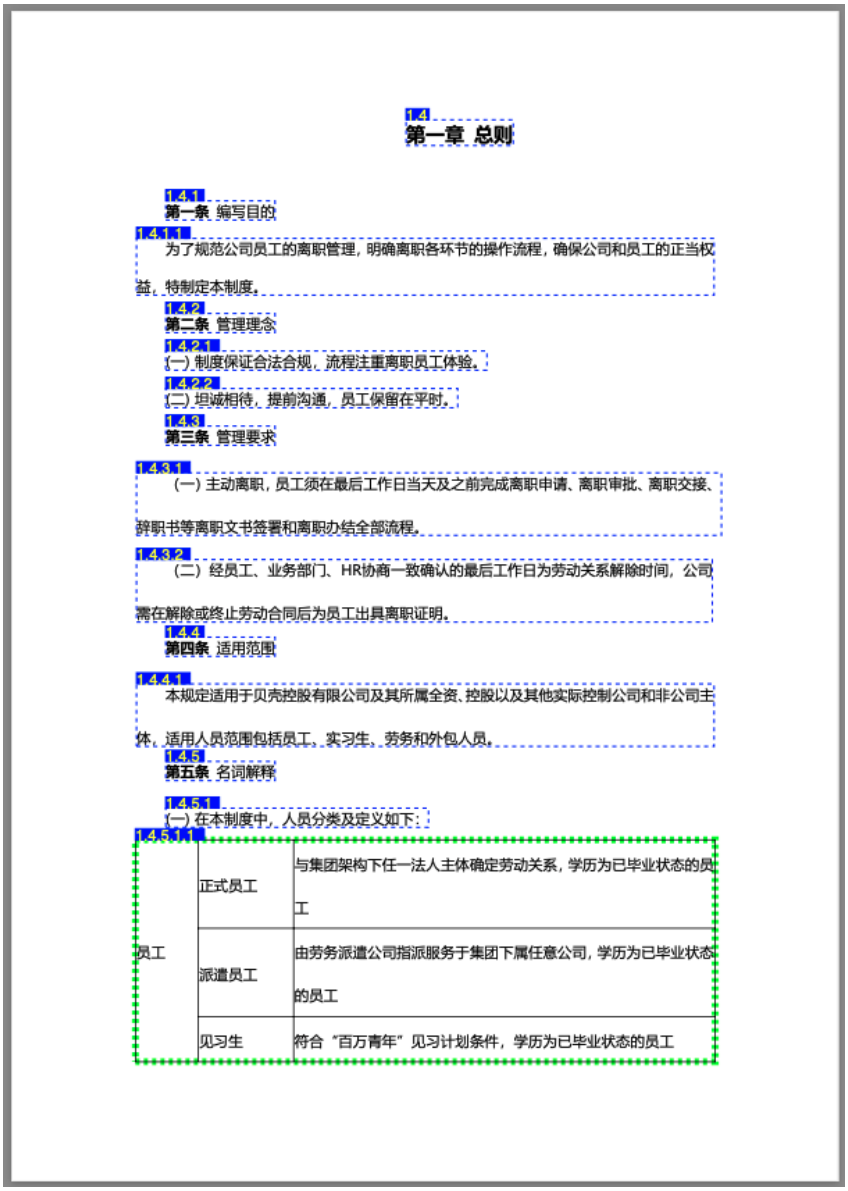
m 为单节点映射为解析节点的个数

3、 层级结构正确率

在完成片段切分和识别后，解析引擎还需具备将片段在文章中的位置进行还原的能力，以构建完整的层级结构，如下图所示。这样，无论哪个文章片段，我们都能清楚地知道它属于哪一章哪一节，以及描述的具体内容。

```
├── PDF 解析：从视觉到结构的重建之旅
    ├── 引言
    ├── PDF 的原理与解析挑战
    ├── 相关工作
    ├── PDF 解析评测
        ├── 构建评测集
        ├── 设计评测指标
            ├── 片段切分准确率
            ├── 版面元素正确率
            ├── 层级结构正确率
        ├── 编写评测脚本
    ├── 解析优化
        ├── 非正文内容的识别
        ├── 文字合并策略的演进
        ├── 版面元素的识别
        ├── 层级结构的识别
    ├── 迭代历程与未来规划
```

在某个页面中正确的层级结构解析效果，应类似下图：



第一章 总则

第一条 编写目的

为了规范公司员工的离职管理，明确离职各环节的操作流程，确保公司和员工的正当权益，特制定本制度。

第二条 管理理念

(一) 制度保证合法合规，流程注重离职员工体验。

(二) 坦诚相待，提前沟通，员工保留在平时。

第三条 管理要求

(一) 主动离职，员工须在最后工作日当天及之前完成离职申请、离职审批、离职交接、辞职书等离职文书签署和离职办结全部流程。

(二) 经员工、业务部门、HR协商一致确认的最后工作日为劳动关系解除时间，公司需在解除或终止劳动合同后为员工出具离职证明。

第四条 适用范围

本规定适用于贝壳控股有限公司及其所属全资、控股以及其他实际控制公司和非公司主体，适用人员范围包括员工、实习生、劳务和外包人员。

第五条 名词解释

(一) 在本制度中，人员分类及定义如下：

员工	正式员工	与集团架构下任一法人主体确定劳动关系，学历为已毕业状态的员工
	派遣员工	由劳务派遣公司指派服务于集团下属任意公司，学历为已毕业状态的员工
	见习生	符合“百万青年”见习计划条件，学历为已毕业状态的员工

我们以真实的父子关系片段对为基准，统计解析引擎构建的关系中被覆盖的片段对的比例，以此作为层级正确率的衡量标准。该指标定义如下：

$$\text{结构准确率} = \frac{\text{正确父子边数}_{parse树}}{\text{所有父子边数}_{label树}}$$

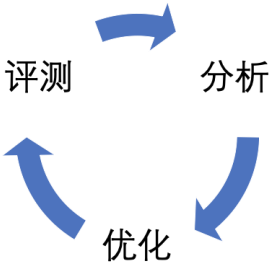
编写评测脚本

在这一阶段，主要任务是编写脚本以获取不同解析引擎的输出结果，并将这些结果与预先定义的标准解析结构进行转换和对比。该步骤实现对解析引擎的自动化评估，确保评测过程的高效性和一致性。

在最初的评测阶段，贝壳自研的 PDF 解析引擎表现相对落后。然而，通过一系列的优化和改进，我们显著提升了引擎的性能，以下是我们采取的一些关键措施。

解析优化

在实现可评测之后，我们迅速确立了“评测-badcase 分析-迭代-再评测”的工作模式。



根据评测指标所反映的解析结果 badcase，分析解析失败的原因，在优化相应的解析策略后，重新进行评测，观测最终是否提升整体解析效果来确定改进方向。

下面我们来看一下，整个解析工作的探索过程。

非正文内容的识别

在项目初期，我们发现文档解析结果中若包含非正文内容（如目录、页眉页脚），会对后续使用造成困扰。例如，目录可能列出“第三章 第四节 交易房屋分类”，但不包含实际内容；页眉页脚则可能在物理上分隔逻辑上相连的文字段落，阻碍文字合并。因此，我们需要识别并去除文章中的非正文内容，包括目录、封面、页眉页脚等。

下面我们以**页眉页脚的识别**为例，看下整个策略的演进过程

阶段一：建立识别页眉能力

【假设一】页眉是每页的第一个文字块，且字符内容大体不变
【策略一】收集每页前 n 个文字块，若文字块去除数字后内容一致，则判定为页眉元素。
【评测效果】可识别大部分页眉，进而提升跨页文本合并效果，block 切分提升 1pp

阶段二：引入坐标信息提升效果

badcase 示例：页眉不止一个文字块，最多甚至超过 10 个文字块

我们发现，不应该只页眉块的数量去做召回，页眉的最明显的特征应该位置信息，必须引入文字块的坐标信息作为主要依据才更准确合理，于是我们继续探索

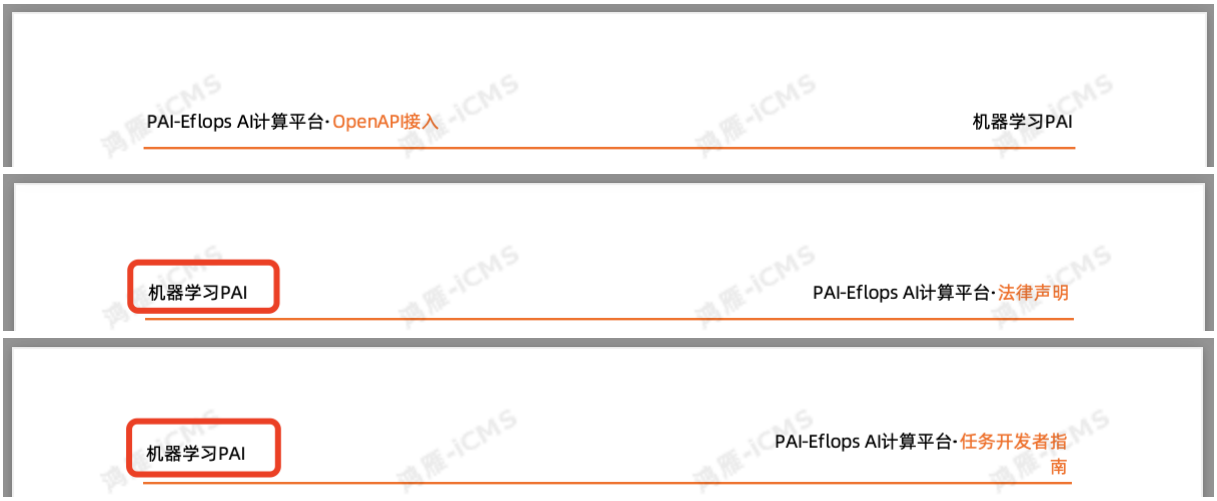
【假设二】页眉集中在整个页面高度的 X%区域，字符内容及所处位置大体不变

【策略二】**引入疑似页眉区和坐标**：每页 X%区域设为疑似页眉区，取其中文字块进行比对，若文字块去除数字后内容一致及**坐标基本不变**，则判定为页眉元素。

【评测效果】可识别图片页眉，且识别不受页眉块的个数影响，block 切分提升 1pp

阶段三：鲁棒性进一步提升

badcase 示例：变化页眉



无论页眉区域内是文字还是图片块，无论文字和图片如何变化，页眉区的内容都应该认定为页眉元素，那么核心在于通过某个细节确定页眉区，再统一认定页眉区元素，于是

【假设三】文档中如存在页眉，无论页眉内容是否有变化，区域是不变的

【策略三】在**疑似页眉区内，**若文字或图片块被认定为页眉元素，则认定元素的底座标以上可判定为**最终页眉区**，所有页面处于最终页眉区的块均被判定为页眉元素。

【评测效果】能识别变化页眉，且识别不受单个页面的内容影响，block 切分提升 1pp

文字合并策略的演进

在页眉与页脚识别问题得到充分解决后，我们聚焦于跨页文本的合并的策略优化中。

根据文字块的样式、位置、字符内容等方式，先粗切分组，再精切确定。

切分策略

【粗切标准】包括

基于缩进：基于行的坐标信息，判断缩进位置，进而判断段首/段尾行作为分组依据；

基于行尾标点：对于冒号“：”、破折号“-”、分号“；”等 17 种特殊符号进行补充判断；

基于对齐方式：同一分组内对齐方式相同，若出现新的对齐方式则划为新组；

等等

【细切标准】包括

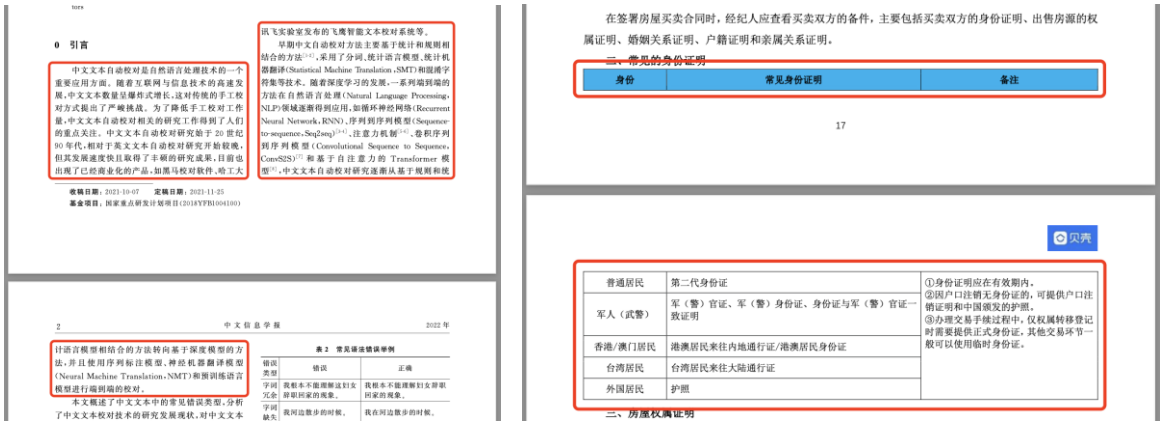
基于样式：上下两行若字体、字号、粗体等样式信息不同，进行切分；

基于间距：基于组内所有行间距，统计行间距众数，进行块切分；

基于有序列表等：对于“(1)”、“①”、“[1]”等有/无序列表头进行识别并切分；

补充策略一：跨页、节、列的文字合并

下图所示的一篇中文论文文档中，可以看到一个段落或一个表格被多页、多列展示时切分，针对该类问题，我们补充了块合并策略。



补充策略二：悬挂缩进

下图所示的参考文献中，常见悬挂缩进+左对齐方式的布局，虽然行首有缩进，但并不是新的文字块，其应与带有序号的上一行同属一个文字块。我们针对该布局进行了处理：通过正则匹配到有序列表的列表头进行缩进长度判断，在精切时保证切分结果准确。

参考文献

[1] 徐连诚，石磊. 自动文字校对动态规划算法的设计与实现[J]. 计算机科学，2002，29(9)：149-150.

[2] 龚小谨，罗振声，骆卫华. 中文文本自动校对中的语法错误检查[J]. 计算机工程与应用，2003，39(8)：98-100.

[3] Cho K，Van Merrienboer B，Gulcehre C，et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[C]//Proceedings of the Conference on Empirical Methods in Natural Language Processing. Stroudsburg，PA，USA：Association for Computational Linguistics，2014：1724-1734.

版面元素的识别

对于文档版面元素的识别能力，我们定义可识别的版面元素如下：

分类	枚举	版面元素含义
区域类	Header	页眉
	Footer	页脚
	Cover	封面
	Catalog	目录
文本类	Title	标题
	List	列表
	Formula	公式
	Code	代码块
	Text	普通文字
图片类	Figure	图片
	FigureName	图名
	FigureNote	图注
表格类	Table	表格
	TableName	表名
	TableNote	表注

以 Title 的识别为例，我们一起看下是如何从 0 到 1 建立起该元素的识别策略的。

第一步：定义 Title

首先我们要定义清什么是 Title：Title 一般指一段文字或者内容的总结性文字，通常用于概括其以下的段落、章节或整个文档的主要内容，一般不会很长、不会跨行；

第二步：分析 Title 特征

- **样式特征：**
    1) 字号大小通常比正文大
    2) 可能使用不同的字体样式
    3) 可能加粗
    4) 颜色可能与正文不同，通常更醒目
- **布局特征**
    1) 可能会居中，可能无缩进左对齐，基本不会出现右对齐、和有缩进的情况
    2) 与其对应正文的行间距可能会更大
- **格式特征**
    1) 可能会使用编号，例如【1 引言】【第一章 序言】【1.3.2 房源维护方法】
    2) 可能包含特定字符，例如：前言、目录、序言、附录、概述等
- **其他特征**
    1）目录中的目录项，基本都可以认定为 Title；
    2）Title 一定会有其对应的正文，一般为段落，也可能为表格、图像；

第三步：设计识别策略

1、目录判定：通过在文档中提取的目录信息，进行初步判定

2、层级判定：通过文字块的子节点数量进行辅助判定

3、格式判定：通过定义的 Title 格式进行判定（例如文字居中且与下文样式不同、行间距有变化等）

当前基于评测集的 Title 的识别率可达到 91.1%

层级结构的识别

层级结构的识别是将整个文章零散的字段整理统一的重要一步，这里我们发现在版面元素识别部分，我们已经掌握了一些层级信息，例如：

- 目录中已将文章的“骨架”部分展示给我们；

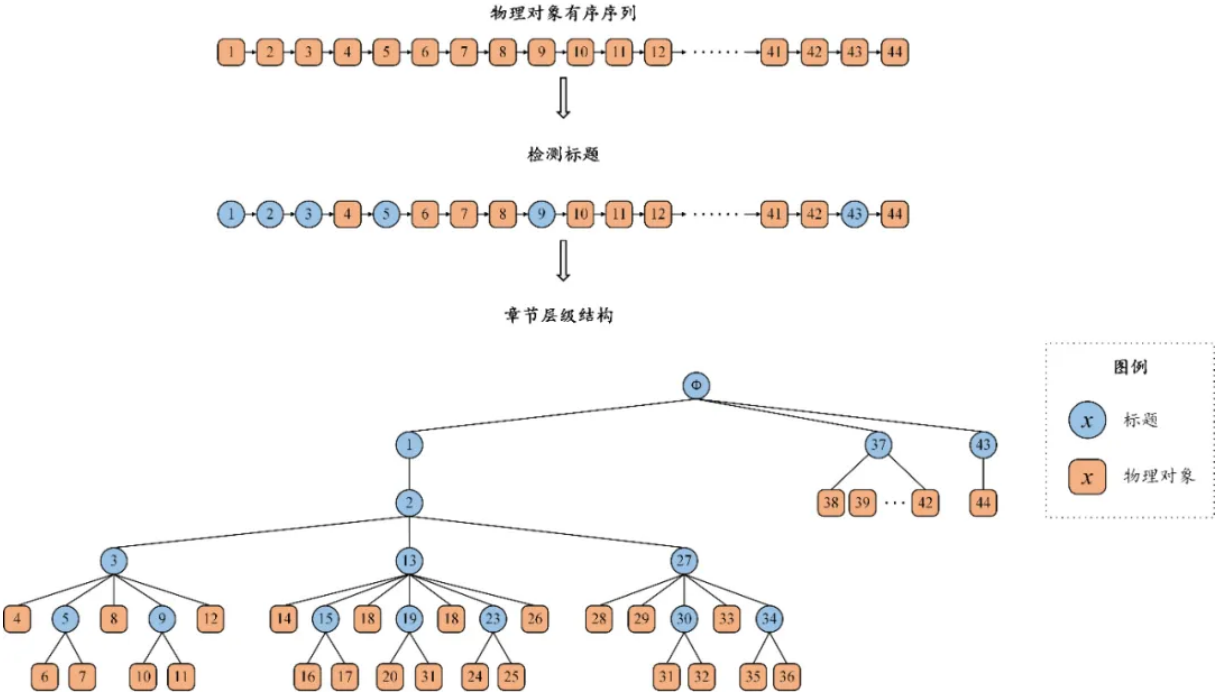
- 在识别列表时，相似的两个列表（例如①xxx ②xxx）应属于同一层级；

在此基础上，再补充一些假设性规则，例如：

- Title 的父节点必须也是 Title；

- 样式相同的临近字块应属于统一层级；

这时，我们就可以将平铺的文字/表格/图像块，进行遍历，通过以上的综合层级关系判断标准进行判断是否能挂载到目标块上即可。流程如下图所示：



迭代历程与未来规划

PDF 解析的整个过程并非一蹴而就，而是一个持续打磨和优化的过程。

从项目初期的不可评测，到可评测但解析效果不佳，再到目前跻身业界顶尖的解析水平，我们前后共经历了六个大版本的优化迭代，建立了“评测-分析-优化”的闭环工作模式，发表了专利一篇。

期间项目的关键发包节点如下：

```
version = '0.1.0.1'  20240717   简单解析接口支持docx、pptx、pdf；
version = '0.1.0.2'  20240731   简单解析接口支持过滤页眉，S3批量上传；
version = '0.1.0.3'  20240815   FAQ 判断 LLM 不稳定问题优化
version = '0.1.1.0'  20240816   页眉新方案上线；
version = '0.1.2.0'  20240822   目录、封面识别并去除；接口解析接口图片附带S3链接；
version = '0.1.3.0'  20240823   封面识别前提大于3页(含)
version = '0.1.3.1'  20240826   parser日志输出优化
version = '0.1.3.2'  20240826   读取config强校验去除
version = '0.1.3.4'  20240829   修复特殊字符引起的解析结果打印异常;页脚阈值放宽；
version = '0.1.3.6'  20240830   不常见字体兼容；
version = '0.1.3.7'  20240912   FAQ文件QA切分输出；
version = '0.1.3.10' 20240919   大文件oom问题优化；
version = '0.1.3.11' 20240919   image_s3_link属性丢失问题修复；
version = '0.1.3.12' 20240920   特殊字体的默认读入设置改为Helvetica；
version = '0.1.3.13' 20240923   list正则补充（1.2.3式）；
version = '0.1.4.0'  20240923   block优化里程碑0.96；
version = '0.1.5.0'  20241022   层级优化里程碑0.80  (336 / 422)；
version = '0.1.5.1'  20241023   表头识别为页眉case优化；
version = '0.1.5.2'  20241023   日志优化；（庖丁评测效果产出）
version = '0.1.5.4'  20241024   不规则目录识别；目录过滤可选；
version = '0.1.5.5'  20241104   页眉阈值优化；（surya评测效果产出）
version = '0.1.6.0'  20241126   block优化（论文解析跨column合并优化、切分锚点改为统计量）；
version = '0.1.7.0'  20241129   工程接口提供，打通FileAPI服务；
```

未来，在 PDF 解析方面，我们希望在解析难点上取得更多突破，包括表格中的三线表、开放式表格，以及图片中的线框图和图像文字识别等。

不仅限于 PDF 文档的解析，我们还计划扩展到更多类型的文档，如 PPTX 和 Wiki，以提升文件的再利用效率。

-- END --