

通用 OCR 大模型引擎 输出协议

版权所有 © 科大讯飞股份有限公司 2024。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

法律声明

本文所描述内容可能包含但不限于对非科大讯飞或开源软件的介绍或引用，使用它们时请遵循对方的版权要求。

注意

您购买的产品、服务或特性等应受科大讯飞公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，科大讯飞公司对本文档内容不做任何明示或暗示的声明或保证。由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目录

目录

1、概述	6
1.1 概述	6
1.2 名词和缩写	6
2、JSON 输出协议	8
2.1 协议概述	8
2.1.1 要素输出顺序与多候选说明	9
2.1.2 要素坐标与角度说明	10
2.1.3 要素属性表示说明	11
2.1.4 要素跨页、跨区域表示说明	11
2.2 协议说明	13
2.2.1 页面	17
2.2.2 版面	23
2.2.3 区域	27
2.2.4 页眉	30
2.2.5 标题	32
2.2.6 段落	35
2.2.7 文本行	37
2.2.8 表格	51
2.2.9 插图	58
2.2.10 列表	60
2.2.11 公式	63
2.2.12 代码	65
2.2.13 伪代码	67
2.2.14 信息栏	69
2.2.15 印章	72
2.2.16 指印	77
2.2.17 条形码	80

2.2.18 二维码	82
2.2.19 水印	84
2.2.20 页脚	86
2.2.21 页码	88
2.2.22 批注	90
2.2.23 脚注	92
2.2.24 键值对	93
2.2.25 目录	100
2.3 特殊场景示例	102
2.3.1 文档与非文档混合场景	103
2.3.2 产品文档场景	105
2.4 常见问题说明	109
2.4.1 表格嵌套表格识别	109
2.4.2 要素输出排序	109
3、Markdown 输出协议	109
3.1 协议概述	109
3.2 协议说明	110
3.2.1 标题	112
3.2.2 段落	112
3.2.3 表格	114
3.2.4 列表	116
3.2.5 代码	117
3.2.6 公式	117
4、SED 输出协议	119
4.1 协议概述	119
4.2 协议说明	119

时间	版本	描述	作者
2024-04-29	1.0	基线版本	程志鹏、陈天宇、韩球
2024-06-17	1.1	JSON 输出结果所有要素节点增加角度（angle）字段	李解
2024-09-19	1.2	修改文本单元属性	刘泽霆
2024-10-10	1.3	1. 增加键值对（key 和 value）节点 2. page 节点中增加 attribute 属性，用于表示医疗场景的图像分类结果	王鹏飞
2024-11-26	1.4	1. 新增要素目录 2. 单元格类别新增“虚拟表头” 3. 修改前景背景颜色的支持范围	刘泽霆
2024-1-10	1.5	1. 文本块属性增加 item_number 2. 公式增加 category 字段，用以区分化学分子式和其他公式 3. 修改要素可包含关系，页眉、页脚可包含页码，标题可包含公式	刘泽霆
2025-2-19	1.6	1. 扩充要素包含关系（详见 2.2 节） 2. 增加签章属性 3. 修改 item 的 content 节点为两级列表，和引擎保持一致	刘泽霆

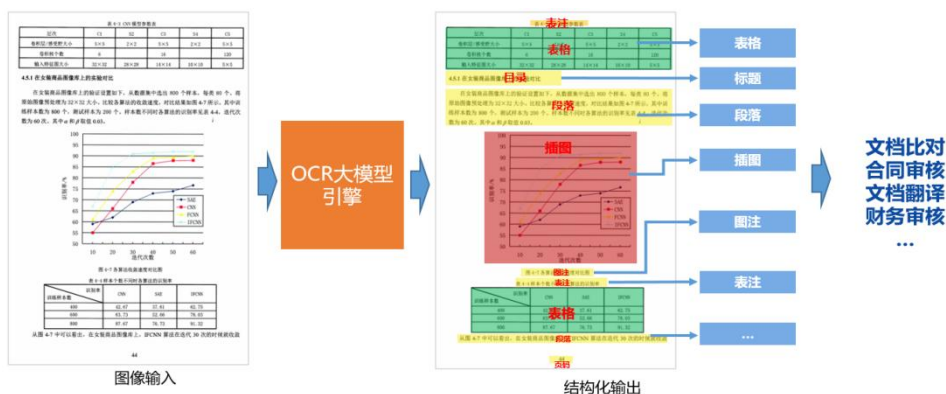
1、概述

1.1 概述

本协议是通用 OCR 大模型引擎输出格式的说明文档，详细描述了识别结果的格式、规范，以及相关属性、参数的类型和取值范围。通用 OCR 大模型引擎支持拍照、扫描、网页截图等各种文档和非文档图像识别，识别结果以 JSON、Markdown 格式输出，本文档针对 JSON、Markdown 格式的输出规范给出了具体说明。



通用 OCR 大模型引擎除了支持文字识别以外，还支持页眉、页脚、页码、表格、插图等要素的结构化识别。



1.2 名词和缩写

以下列出了文档中出现的一些名词和缩略语：

(1) 要素：是 OCR 识别结果的基本组织单位，用于表示图像上识别结果的组织关系。要素有 25 类，分别是页面、版面、区域、段落、文本行、页眉、标题、印章、指印、二维码、信息栏、水印、条形码、页码、表格、插图、列表、代码、伪代码、脚注、公式、批注、页脚、键值对、目录。

(2) 固定要素：在图像上的位置是相对固定的。其中页面、版面、区域、段落、文本行、页眉、标题、信息栏、页码、表格、插图、列表、代码、伪代码、脚注、公式、批注、页脚、目录是固定要素。

(3) 浮动要素：在图像上的位置是不固定的，其中二维码、条形码、水印、指印、印章是浮动要素。

(4) 文档图像：是指图像里含有公文、合同、书籍、杂志、票证等版式固定的文档。例如打印 Word、Excel 文件得到的图像，拍照合同、报告得到的图像等。

(5) 非文档图像：与文档图像相反，是指图像里不含有固定版式的文档，例如用手机拍照的广告牌、路标、商品标签等图像。

(6) SED 格式：是 Simple Element Document 的简称，表示采用简单要素格式输出识别结果，采用 SED 格式时，识别的结果仅以段落和表格的形式进行组织，其他要素均转换为此两种格式表示。

2、JSON 输出协议

2.1 协议概述

JSON 输出协议明确了输出结果的表示规范，协议说明部分描述了各种要素的定义、属性和表示规范等。针对特殊和复杂场景，为了便于读者理解，给出了具体示例。下面给出一个输出结果 JSON 的简单示例，帮助读者对 JSON 输出协议有一个直观的了解。

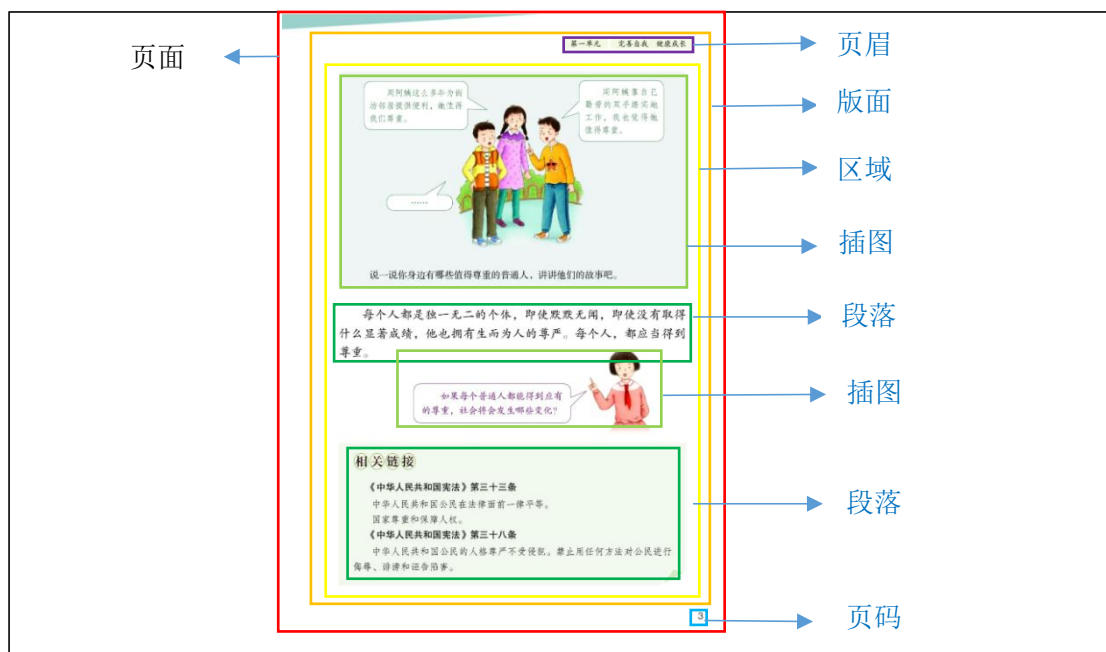


图 2-1-1：要素示例图

上图中间是一张图像，经过引擎结构化识别后得到各种要素，要素位置用红、绿、蓝等颜色的边框标记在图像上，要素名称位于图像两侧。除了指印要素以外，其他要素均可以包含子要素，例如上图中的插图要素，可以包含段落子要素；子要素均存放在 `content` 节点下，上图对应的 JSON 如下表所示，为了便于说明，省略了部分字段。

```
{
  "engine_version": "1.0.0.1001",           //引擎版本号
  "document": [],                             //输出文档类型
  "version": "1.0",                          //输出协议版本号
  "image":
  [
    {
      "id": "xxx",                           // 图片 id
      "width": 2400,                         // 图片宽度
      "attribute": [],                       // 图片属性
      "height": 2200,                       // 图片高度
      "content":
      [
```

```
[
  {
    "type": "page",           // 页面
    "content":
    [
      [
        {
          "type": "page_header" // 页眉
        },
        {
          "type": "layout",     // 版面
          "content":
          [
            [
              {
                "type": "region", // 区域
                "content":
                [
                  [
                    {
                      "type": "graph" // 插图
                    },
                    {
                      "type": "paragraph" // 段落
                    },
                    {
                      "type": "graph" // 插图
                    },
                    {
                      "type": "paragraph" // 段落
                    }
                  ]
                ]
              }
            ]
          ]
        },
        ...
      ],
      {
        "type": "page_number" // 页码
      }
    ]
  }
]
```

2.1.1 要素输出顺序与多候选说明

通用 OCR 大模型引擎识别的各类要素均保存在 `content` 节点下，`content` 节点是一个二维数组，其中数组的第一个维度表示多候选，第二个维度表示子要素集合。多候选按照得分（`score`）从高到底进行排序，得分最高的保存在数组的第一项，得分次高的存放在数组的第二项，以此类推；子要素按照阅读顺序保存在 `content` 数组的第二个维度中，即数组的第二个维度是按照阅读顺序排序的。

下面给出一个示例：

```
{
  "engine_version" : "1.0.0.1001",           //引擎版本号
  "document" : [],                           //输出文档类型
  "version": "1.0",                          //输出协议版本号
  "image" :                                  //每个图片的输出结果
  [
    {
      "id": "1",                             // 图片 id
      "width": 2400,                         // 图片宽度
      "height": 2200,                       // 图片高度
      "attribute": [],                      // 图片属性
      "content":                             // 图片要素内容
      [
        // 多候选 top1
        [
          {
            "id": "11",
            "type": "title",                //标题要素
            ...
          },
          {
            "id": "12",
            "type": "paragraph",           //段落要素
            ...
          },
          ...
        ],
        // 多候选 top2
        [
          {
            "id": "21",
            "type": "title",                //标题要素
            ...
          },
          {
            "id": "22",
            "type": "paragraph",           //段落要素
            ...
          },
          ...
        ]
      ]
    }
  ]
}
```

2.1.2 要素坐标与角度说明

通用 OCR 大模型引擎支持输出要素的坐标，坐标（coord）由 4 个点组成，采用矩形坐标系计算位置，坐标以 Y 轴方向值最大（最下方）的点为 p[0]（Y 轴坐标相等时取两者中 X 轴坐标大的），依次顺时针标记 p[1]，p[2]，p[3]。坐标点 X 轴取值范围[0，图像宽度 - 1)，坐标点 Y 轴取值范围[0，图像高度 - 1)。

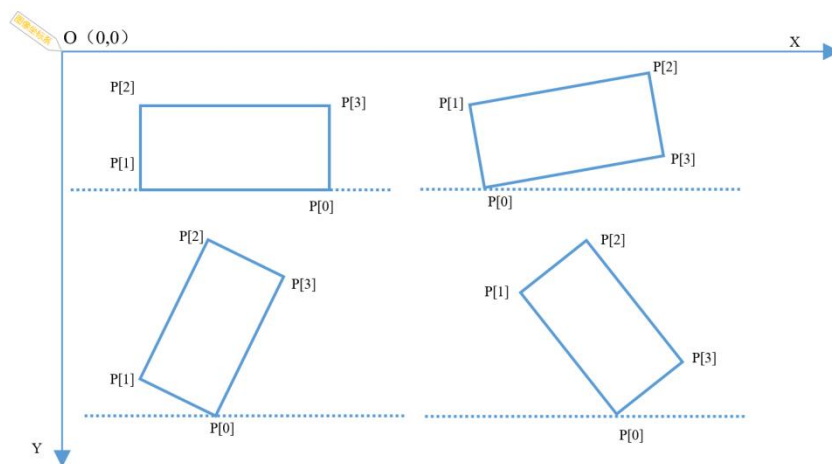


图 2-1-2-1：坐标说明

通用 OCR 大模型引擎支持输出要素角度信息，角度（angle）为 X 轴逆时针旋转到文字阅读方向的角度，取值范围为[0, 360) 度。

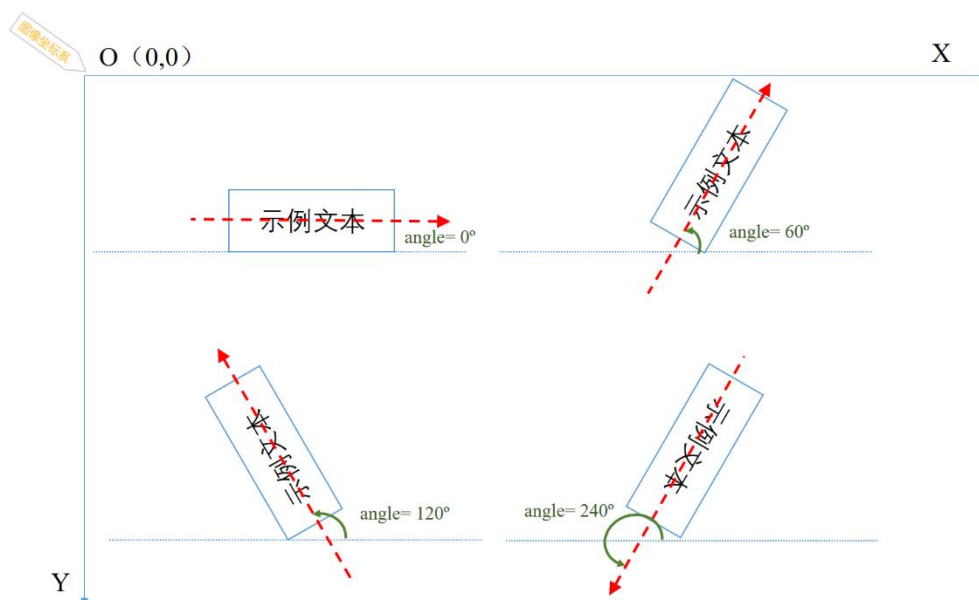


图 2-1-2-2：角度说明

2.1.3 要素属性表示说明

通用 OCR 大模型引擎支持输出要素的属性，属性（attribute）是一个一维数组，由多个节点组成，节点包含 name、value 字段，本节重点介绍文本单元（text_unit）要素的属性。

文本单元的属性节点包含 **name**、**value** 字段，**name** 字段表示属性的名称，例如背景色（**background_color**）、模糊（**ambiguous**）等；**value** 字段表示对属性的描述信息，例如背景色的颜色。

下面给出一个简单示例

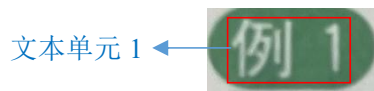


图 2-1-3-1：属性示例

上图中，文本单元 1 “例 1” 具有背景色属性，属性（**attribute**）节点 **name** 取值为 **background_color**，**value** 取值为绿色（**#008000**）。上图对应的 JSON 结果如下：

```
{
  // 文本单元 1
  {
    "type": "text_unit",
    ...,
    "category": "print",
    "attribute":
    [
      {
        "name": "background_color",
        "value": "#008000"
      }
    ],
    "text": "例 1"
  }
}
```

文本单元部分属性只具有 **name** 字段，比如粗体（**bold**）、斜体（**italic**）等，具体示例见 2.2.7.6.2 节。

2.1.4 要素跨页、跨区域表示说明

实际场景中，存在要素跨页、跨栏的情况，例如一个段落可能跨越两个或者多个区域（栏目），每个区域均包含一个子段落，为了既可以表示版面的固有结构，又能保持读取内容的连贯性，引入了关联属性（**relation**），表示相同子要素之间的关系，关联属性存放在属性（**attribute**）节点下。当存在跨页、跨区域等要素时，每个子要素属性节点下增加名为 **relation** 的项目，该项目的 **value** 字段按照阅读顺序存放所有子节点编号（**id**），这样每个子要素之间建立了联系。存在跨页、跨区域表示的要素有：区域、段落、表格、插图、代码、伪代码、列表、目录。

2.2 协议说明

JSON 协议的根节点有以下字段：

字段名称	字段类型	是否必须	说明
engine_version	String	是	引擎版本号
version	String	是	协议版本号
image	Image[]	是	图像集合（大小与输入图像数相同）
document	Object[]	是	输出的格式文档，name是文档类型，value是文档对应的字符串数据。例如： <pre>{ "name": "word", "value": "xxx" }, { "name": "markdown", "value": "xxx" } { "name": "sed", "value": [] }</pre> name取值如下： markdown: Markdown格式，具体格式见第三章Markdown输出协议 sed: 简单要素格式（仅包含纯文本、markdown格式表格），具体具体格式见第四章SED输出协议 word: Word docx格式 excel: Excel xlsx格式

上表中，image 字段用于存放输入图像的识别结果，image 是一个数组，数组的大小与输入图像的张数对应，其中第 1 张图像对应 image 下标为 0 的节点，第 2 张图像对应 image 下标为 1 的节点，以此类推。引擎目前仅支持输入单张图像进行识别。图像（image）节点字段说明如下：

image			
字段名称	字段类型	是否必须	说明
id	String	是	编号

width	Int	是	图片宽度
height	Int	是	图片高度
attribute	Object[]	是	图片的属性集合
content	Object[][]	否	子要素集合。其中数组的第一个维度表示多候选，第二个维度表示子节点集合
angle	Float	是	v1.1版本新增： 图片旋转角度，取值范围[0, 360)， 图片的所有子要素的坐标（coord）、 轮廓（contour）均为图片绕中心点顺 时针旋转一定角度后的图片上对应的 坐标、轮廓。

图像的属性（attribute）集合字段说明：

属性	字段名	字段类型	字段说明
拒识	name	String	rejection
	value	Int	-1表示识别异常

下图给出了一个多候选输出示例：



图 2-2-1：图像节点多候选示例

上图对应的 JSON 结果如下：

```
{
  "engine_version" : "1.0.0.1001",           //引擎版本号
  "document" : [],                           //输出文档类型
  "version": "1.0",                          //输出协议版本号
  "image" :                                  //每个图片的输出结果
  [
    {
      "id": "XXX",                           // 图片 id
      "width": 2400,                         // 图片宽度
      "attribute": [],                       // 图片属性
      "height": 2200,                       // 图片高度
      "content":                             // 图片要素内容
      [
        // 多候选 top1
        [
          {
            "text": ["道德与法治"]
          }
        ]
      ]
    }
  ]
}
```

```
    }  
  ],  
  // 多候选 top2  
  [  
    {  
      "text": ["首德与法治"]  
    }  
  ],  
  // 多候选 top3  
  [  
    {  
      "text": ["首德与去治"]  
    }  
  ]  
]  
}
```

`content` 存放了识别结果的要素集合，引擎支持标题、页眉、页脚等 23 种要素，除了指印要素以外，其他要素均支持包含子要素，以下给出每个要素的具体说明：

要素名	要素简称	说明
页面	page	固定要素，仅可作为子要素存在于image要素内，page子要素仅可包括浮动要素、page_header要素、page_number要素、page_footer要素、information_bar要素、layout要素
版面	layout	固定要素，仅可作为子要素存在于page要素内，layout子要素仅可包括region、information_bar要素
区域	region	固定要素，仅可作为子要素存在于layout要素、图注、表注、公式注解内，region子要素仅可包含region、title、paragraph、table、list、pseudocode、code、footnote、formula、graph、annotation、contents、qrcode、barcode、key、value要素
页眉	page_header	固定要素。仅可作为子要素存在于page要素中，page_header子要素仅可包括paragraph、table、graph、page_number、formula、list、qrcode、barcode、key、value要素
标题	title	固定要素，仅可作为子要素存在于region、graph、cell、contents、information_bar要素内，title子要素仅可包括paragraph、graph、formula要素
段落	paragraph	固定要素，仅可作为子要素存在于region、graph、table中单元格cell、title、page_header、page_footer、page_number、watermark、barcode、qrcode、seal、information_bar、list中item、pseudocode、code、footnote、formula、key、value、contents、annotation要素内，paragraph子要素仅可包含textline要素
文本行	textline	固定要素，仅可作为子要素存在于paragraph要素内，textline子要素仅可包含text_block要素

表格	table	固定要素， 仅可作为子要素存在于 region、 graph、 information_bar、 page_header、 page_footer、 footnote、 cell、 annotation要素内， table子要素仅可包含单元格cell， cell子要素仅可包含paragraph、 graph、 list、 code、 pseudocode、 formula、 contents、 title、 qrcode、 barcode、 key、 value、 table要素； 单元格中允许嵌套表格；
插图	graph	固定要素， 仅可作为子要素存在于information_bar、 watermark、 qrcode、 barcode、 page_header、 page_footer、 page_number、 title、 region、 cell、 footnote、 item、 key、 value要素内， 插图子要素仅可包含paragraph、 table、 graph、 contents、 formula、 title、 list、 qrcode、 barcode、 code、 pseudocode、 key、 value要素
列表	list	固定要素， 仅可作为子要素存在于region、 graph、 cell、 page_header、 page_footer、 footnote、 value、 information_bar要素内， 列表子要素仅可包含列表项item， item子要素仅可包含列表list、 paragraph、 graph、 formula要素
公式	formula	固定要素， 仅可作为子要素存在于region、 key、 value、 cell、 graph、 pseudocode、 footnote、 information_bar、 page_header、 page_footer、 title、 annotation、 watermark、 item要素内， formula子要素仅可包含paragraph要素
代码	code	固定要素， 仅可作为子要素存在于region、 cell、 graph、 information_bar要素内， code子要素仅可包含paragraph要素
伪代码段	pseudocode	固定要素， 仅可作为子要素存在于region、 cell、 graph、 information_bar要素内， pseudocode子要素仅可包含paragraph、 formula要素
信息栏	information_bar	固定要素， 仅可作为子要素存在于page要素中， information_bar子要素仅可包括paragraph、 table、 graph、 title、 code、 contents、 pseudocode、 list、 key、 value、 formula、 annotation、 barcode、 qrcode要素
印章	seal	浮动要素， 仅可作为子要素存在于page要素中， seal子要素仅可包括paragraph、 formula要素
指印	fingerprint	浮动要素， 仅可作为子要素存在于page要素中， fingerprint无子要素
条形码	barcode	固定要素， 仅可作为子要素存在于region、 cell、 graph、 page_header、 page_footer、 information_bar、 footnote 要素中， barcode子要素仅可包括paragraph、 graph要素
二维码	qrcode	固定要素， 仅可作为子要素存在于region、 cell、 graph、 page_header、 page_footer、 information_bar、 footnote要素中， qrcode子要素仅可包括paragraph、 graph要素
水印	watermark	浮动要素， 仅可作为子要素存在于page要素中， watermark子要素仅可包括paragraph、 graph、 formula要素
页脚	page_footer	固定要素， 仅可作为子要素存在于page要素中， page_footer子要素仅可包括paragraph、 table、 graph、 page_number、 formula、 list、 qrcode、 barcode、 key、 value要素

页码	page_number	固定要素，仅可作为子要素存在于page、page_footer、page_header要素中，page_number子要素仅可包括paragraph、graph、formula要素
批注	annotation	固定要素，仅可作为子要素存在于region、information_bar要素内，annotation子要素仅可包含paragraph、formula、table要素（目前暂不支持）
脚注	footnote	固定要素，仅可作为子要素存在于region要素内，footnote子要素仅可包含paragraph、graph、formula、list、table、key、value、barcode、qrcode要素
键值对	key和value	固定要素，仅可作为子要素存在于region、cell、footnote、page_header、page_footer、information_bar、graph要素内，key和value子要素仅可包含paragraph、formula、graph、list要素
目录	contents	固定要素，仅可作为子要素存在于region、graph、cell、information_bar、要素内，目录子要素仅可包含paragraph、formula、graph、title要素。

上表给出了各个要素的说明，其中浮动要素、固定要素的具体解释见 [1.2](#) 节。

2.2.1 页面

2.2.1.1 结构说明

页面（page）是图像中相对独立的一页文字，可以是文档区域，也可以是非文档区域，还可以是两种的混合。文档与非文档具体定义见 [1.2](#) 节名词与缩写说明。页面节点字段说明如下：

page			
字段名称	字段类型	是否必须	说明
id	String	是	编号
type	String	是	类型，填写page，表示页面要素
angle	Float	是	角度，取值范围[0, 360)，按照顺时针方向
category	String	是	文档类别，有： document: 文档 other: 非文档
attribute	Object[]	否	页面的属性集合，可以设置图片分类的结果
score	Float	是	得分，取值范围[0, 1]
coord	Point[]	是	页面的位置，是页面对应区域的最大外接矩形坐标

contour	Point[]	是	页面的轮廓，是页面对应区域的轮廓坐标
image_data	String	否	页面所在区域的图片base64编码数据
content	Object[][]	是	页面内子节点要素集合，子节点要素仅可包括浮动要素、页眉（page_header）、信息栏（information_bar）、页脚（page_footer）、页码（page_number）、版面（layout）。 数组的第一个维度表示多候选，第二个维度表示子节点集合

页面的属性（attribute）集合字段说明：

属性	字段名	字段类型	字段说明
关联要素	name	String	classification
	value	String	图像分类结果，目前仅支持医疗场景

图像分类结果字段说明：

类别	说明
medicine_packaging	药品包装盒
medicine_package_insert	药品说明书
medical_examination_form	医疗检查单
medical_laboratory_report	医疗检验单
discharge_summary	出院小结
outpatient_medical_history	门诊病历
electrocardiogram_medical_examination_form	心电图报告单
health_examination_form	体检表
document_image	文档
medical_image	医学影像
eye_medical_examination_form	眼部检查单
general_image	通用图像
portrait_photo	人像
human_skin_photo	人体皮肤
human_wound_photo	人体伤口
human_tongue_photo	人体舌部
human_head_photo	人体头部
human_eye_photo	人体眼部
body_privacy_photo	人体私处
human_body_specimen_photo	人体标本
medical_prescription	处方单

unknown

未知类型

需要说明是，当一个文档位于图像的内部区域，那么文档区域定义为类别为文档（document）的页面，文档区域以外的内容定义为类别为非文档（other）的页面。下图给出了一个文档与非文档页面示例：

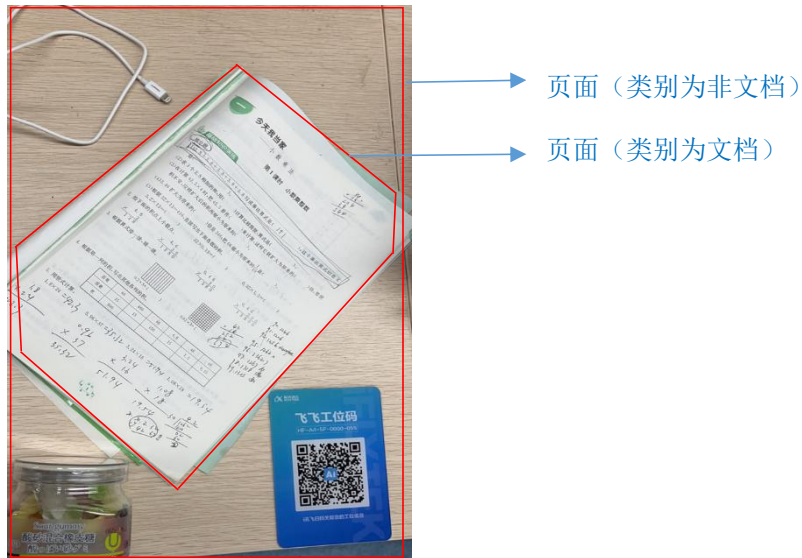


图 2-2-1-1-1：混合场景页面示例

上图对应的 JSON 结果如下：

```
{
  "engine_version" : "1.0.0.1001",
  "document" : [],
  "version" : "1.0",
  "image" :
  [
    {
      "id": "XXX",
      "width": XXX,
      "attribute": [],
      "height": XXX,
      "content":
      [
        [
          // 非文档页面
          {
            "type": "page",
            "category": "other",
            ...
          },
          // 文档场景页面
          {
            "type": "page",
            "category": "document",
            ...
          }
        ]
      ]
    }
  ]
}
```

图像可以没有文档区域，当没有文档区域时，则输出的结果归属于一个类别为非文档（other）的页面。下图是一个没有文档区域的图像：



页面（类别为非文档）

图 2-2-1-1-2：非文档页面示例

上图对应的 JSON 结果如下：

```
{
  "engine_version": "1.0.0.1001",
  "document": [],
  "version": "1.0",
  "image": [
    {
      "id": "1",
      "width": 200,
      "attribute": [],
      "height": 300,
      "content": [
        [
          // 非文档页面
          {
            "type": "page",
            "category": "other",
            ...
          }
        ]
      ]
    }
  ]
}
```

图像也可以仅包含文档区域，此时的输出结果归属于一个类别为文档（document）的页面。下图是一个仅包含文档区域的图像：

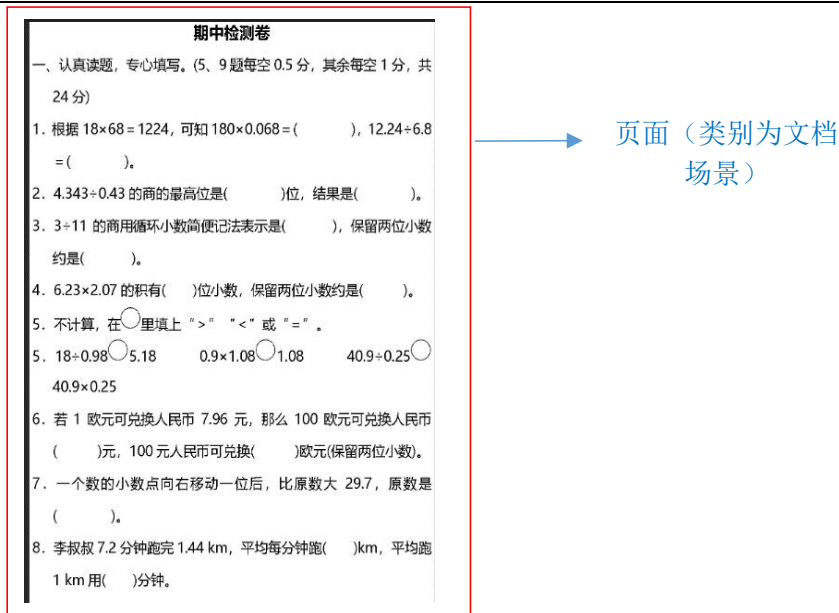


图 2-2-1-1-3：文档场景页面示例

上图对应的 JSON 结果如下：

```
{
  "engine_version" : "1.0.0.1001",
  "document" : [],
  "version" : "1.0",
  "image" :
  [
    {
      "id": "XXX",
      "width": XXX,
      "attribute": [],
      "height": XXX,
      "content":
      [
        [
          // 文档场景页面
          {
            "type": "page",
            "category": "document",
            ...
          }
        ]
      ]
    }
  ]
}
```

2.2.1.2 页面示例

图像可以包括多个页面，下图给出的图像示例包含两个页面：

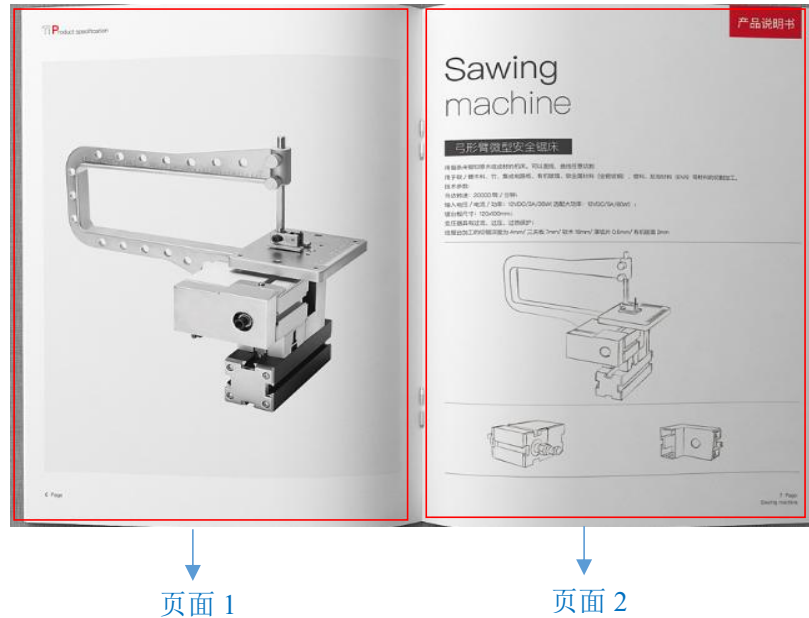


图 2-2-1-2-1：页面示例

上图对应的 JSON 结果如下：

```
{
  "engine_version": "1.0.0.1001",
  "version": "1.0",
  "document": [],
  "image": [
    {
      "id": "XXX",
      "width": 2400,
      "attribute": [],
      "height": 2200,
      "content": [
        [
          // 页面 1
          {
            "type": "page",
            "category": "document",
            "content": [
              // 多候选 top1
              [
                {
                  "type": "layout",
                  ...
                }
              ],
              // 多候选 top2
              [...],
              // 多候选 top3
              [...]
            ]
          },
          ...
        ]
      ]
    }
  ]
}
```

```
// 页面 2
{
  "type": "page",
  "category": "document",
  ...,
  "content":
  [
    [
      {
        "type": "layout",
        ...
      }
    ]
  ]
}
```

2.2.2 版面

2.2.2.1 结构说明

版面（layout）用于表示页面的排版信息，例如论文、报纸、海报等。版面节点的字段说明如下：

layout			
字段名称	字段类型	是否必须	说明
id	String	是	编号
type	String	是	类型，填写layout，表示版面要素
angle	Float	是	v1.1版本新增： 角度，取值范围[0, 360)，按照顺时针方向
coord	Point[]	是	版面的位置，是版面对应区域的最大外接矩形坐标
contour	Point[]	是	版面的轮廓，是版面对应区域的轮廓多边形坐标
content	Object[][]	是	子要素集合，子要素仅可包括区域（region）、信息栏（information_bar）。其中，数组的第一个维度表示多候选，第二个维度表示子节点集合

海报场景示例：



版面

图 2-2-2-1-1：海报场景示例

报纸场景示例：

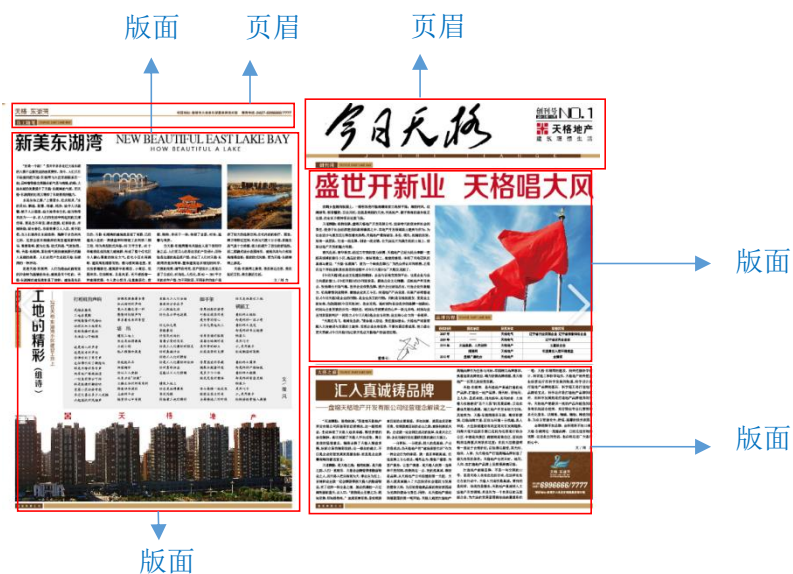


图 2-2-2-1-2：报纸示例

2.2.2.2 简单版面示例

下图给出一个简单版面示例：

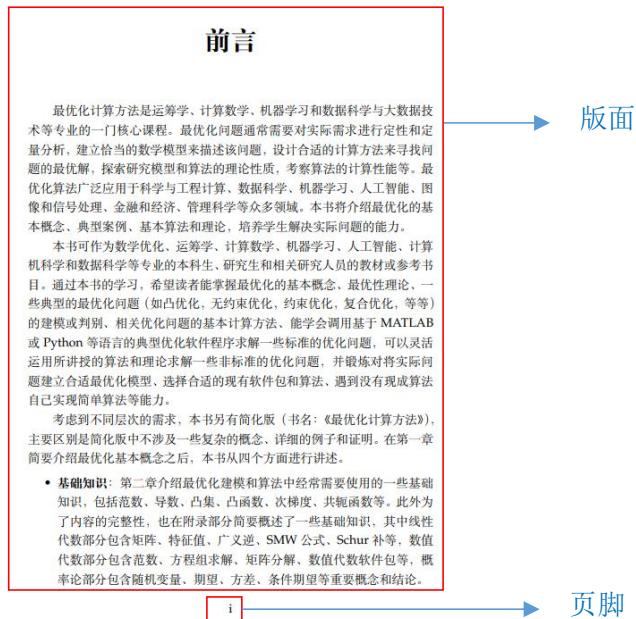


图 2-2-2-2-1：简单版面示例

上图包含一个版面要素和一个与版面同级的页脚要素。版面下的子要素包括标题、段落要素，为了便于说明，未画出版面下的子要素。上图对应 JSON 结果如下：

```
{
  ...
  {
    "type": "page",
    ...
    "content":
    [
      // 版面
      {
        "id": "xxx",
        "type": "layout",
        ...
        "content":
        [
          // 多候选 top1
          [
            {
              "type": "region",
              ...
            }
          ],
          // 多候选 top2
          [...]
        ]
      }
    ]
  }
}
```

2.2.2.3 复杂版面示例

下图给出一个报纸场景的示例：



图 2-2-2-3-1：复杂版面示例

上图对应 JSON 结果如下：

```
{
  "type": "page",
  "content":
  [
    // 版面 1
    {
      "type": "layout",
      "content":
      [
        // 多候选 top1
        [
          {
            "type": "region",
            ...
          }
        ],
        // 多候选 top2
        [
          {
            "type": "region",
            ...
          }
        ]
      ]
    },
    // 版面 2
    {
      "type": "layout",
      "content":
      [
        [
          {
            "type": "region",
            ...
          }
        ]
      ]
    }
  ]
}
```



2.2.3 区域

2.2.3.1 结构说明

区域（region）是页面的排版与布局设计，区域可以包含段落、插图等子要素，还可以包含子区域，区域节点的字段说明如下：

region			
字段名称	字段类型	是否必须	说明
id	String	是	编号
type	String	是	类型，填写region，表示区域要素
angle	Float	是	v1.1版本新增： 角度，取值范围[0, 360)，按照顺时针方向
category	String	是	区域类别： column: 栏目 note: 插图、表格、行间公式的标题或注释
attribute	Object[]	否	区域的属性集合
coord	Point[]	是	区域的位置，是区域的最大外接矩形坐标
contour	Point[]	是	区域的轮廓，是区域的轮廓坐标
content	Object[][]	是	子要素集合，子要素仅可包含区域（region）、标题（title）、段落（paragraph）、表格（table）、列表（list）、伪代码（pseudocode）、代码（code）、脚注（footnote）、公式（formula）、插图（graph）、批注（annotation）、键值对（key和value）、目录(contents)、二维码（qrcode）、条形码（barcode）要素。 其中，第一个维度表示多候选，第二个维度表示子节点集合

以下是区域要素的示例：



图 2-2-3-1-1：区域示例

2.2.3.2 区域类别

区域分为多种类别，在通用场景中，区域分为栏目（column）和注释（note）类别；区域之间可以相互嵌套。

下图给出了一个包含区域的图片示例。示例图像由左右两个区域要素组成，这两个区域的类别均为栏目（column）。



图 2-2-3-2-1：通用场景区域示例

2.2.3.3 区域属性

通用场景中，区域属性（attribute）字段取值说明如下：

属性	字段名	字段类型	字段说明
关联要素	name	String	relation
	value	String[]	存放属于同一区域要素的编号（id）

当区域存在跨栏跨页情况时，每个区域都会赋予关联要素（relation）属性，属性的 value 字段取值为属于同一区域的 id 集合。

2.2.3.4 区域示例

下面给出了一个包含区域要素的图片示例：

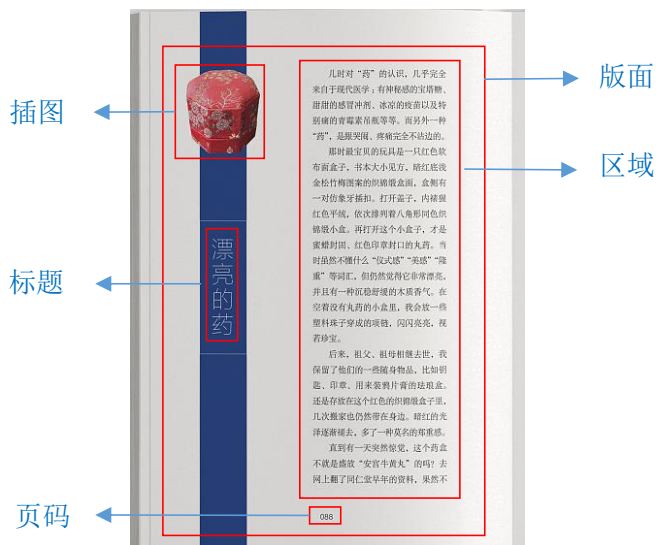


图 2-2-3-4-1：区域示例

上图中，版面要素内包含区域、插图、标题、页码要素，其中区域要素的类别为栏目（column），区域内包含多个段落要素，为了便于说明，图中段落要素未标出。上图对应的 JSON 结果如下：

```
{
  "type": "layout",
  "...": "...",
  "content": [
    [
      // 区域（类别为栏目）
      {
```

```
"type": "region",
"category": "column",
"attribute": [],
"content":
[
  [
    {
      "type": "paragraph",
      ...
    },
    {
      "type": "paragraph",
      ...
    }
  ]
]
}
```

2.2.4 页眉

2.2.4.1 结构说明

页眉（page_header）是指位于文档图像的顶端区域，用于对文档的补充介绍和说明，例如文档编号、图标、日期等信息。页眉仅可作为子要素存在于页面（page）要素中，页眉节点字段说明如下：

page_header			
字段名称	字段类型	是否必须	说明
id	String	是	编号
type	String	是	类型，填写page_header，表示页眉要素
angle	Float	是	v1.1版本新增： 角度，取值范围[0, 360)，按照顺时针方向
coord	Point[]	是	页眉的位置，是页眉对应区域的最大外接矩形坐标
contour	Point[]	是	页眉的轮廓，是页眉对应区域的轮廓坐标
content	Object[][]	是	子要素集合，子要素仅可包含插图（graph）、段落（paragraph）、表格（table）、页码（page_number）、公式（formula）、列表（list）、键值对（key和value）、二维码（qrcode）、条形码（barcode）要素。其中，数组的第一个维度表示多候选，第二个维

度表示子节点集合。

下图给出了一个包含页眉的示例：



图 2-2-4-1-1：页眉示例

上图中，页面包含页眉、版面、标题等要素，为了便于说明，图中仅标注出页眉要素。

2.2.4.2 页眉示例

下图给出了一个包含页眉的图片示例：



图 2-2-4-2-1：页眉示例

上图中，页面内包含页眉和版面要素，页眉包含段落要素，版面内包含标题、段落等要素，为了便于说明，图中仅标出页眉要素，上图对应的 JSON 结果如下：

```
{
  "id": "1",
  "type": "page",
  "category": "document",
  "content":
  [
    [
      {
        "id": "2",
        "type": "page_header",
        "coord": [...],
```

```
"contour": [...],
"content":
[
  [
    {
      "id": "3",
      "type": "paragraph",
      "coord": [...],
      "contour": [...],
      "attribute": [...],
      "content": [...]
    },
    {
      "id": "4",
      "type": "paragraph",
      "coord": [...],
      "contour": [...],
      "attribute": [...],
      "content": [...]
    }
  ],
  [...],
  [...]
]
}
```

2.2.5 标题

2.2.5.1 结构说明

标题（title）是标明文章、作品、段落等内容的简短语句，有助于读者快速理解文章或作品的主题和内容。标题由一行或者多行文字组成，标题节点字段说明如下：

title			
字段名称	字段类型	是否必须	说明
id	String	是	标识符（全局唯一）
type	String	是	类型，填写title，表示标题要素
angle	Float	是	v1.1版本新增： 角度，取值范围[0, 360)，按照顺时针方向
coord	Point[]	是	标题的位置，是标题对应区域的最大外接矩形坐标
contour	Point[]	是	标题的轮廓，是标题对应区域的轮廓坐标
content	Object[][]	是	子要素集合，子要素仅可包含段落（paragraph）、插图（graph）、公式（formula）要素。 其中，数组的第一个维度表示多候选，第二个维度表示子节点内容。

level	Int	是	1-5，对应1-5级标题
-------	-----	---	--------------

下面给出一个包含标题要素的图片示例：

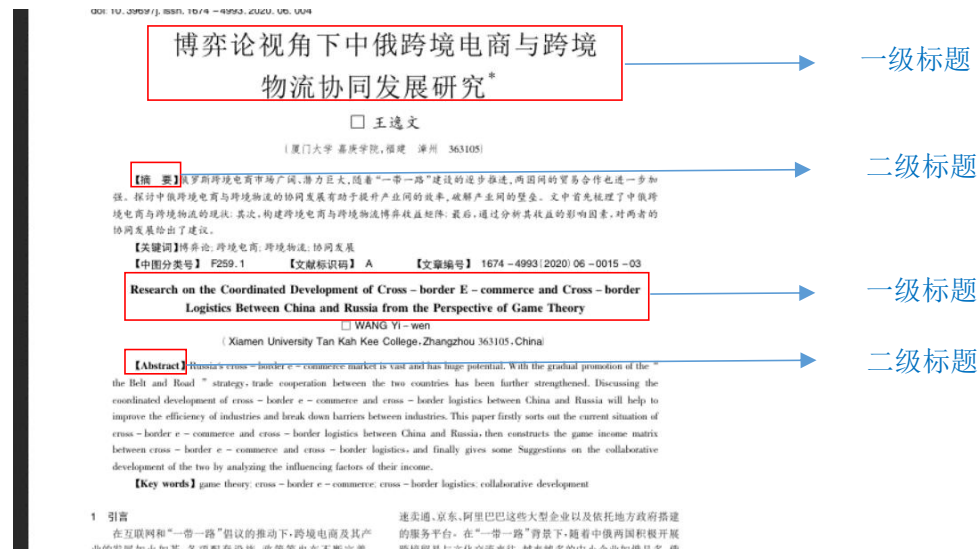


图 2-2-5-1-1：标题示例

下面给出一个包含 1 到 5 级标题的图片示例：

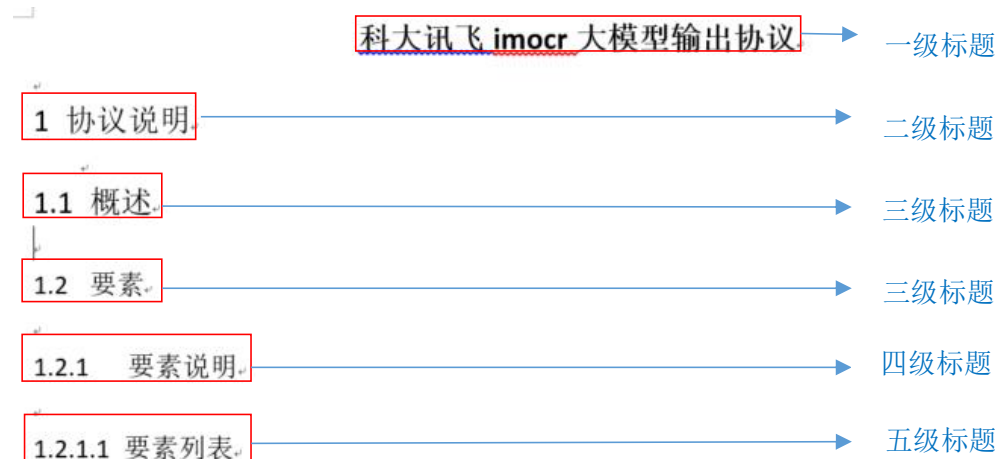


图 2-2-5-1-2：标题等级划分示例

一级标题一般是指文档正上方加粗居中的文字，需要说明的是标题等级并非从第一级开始，可以从任一级别开始，如下图所示：

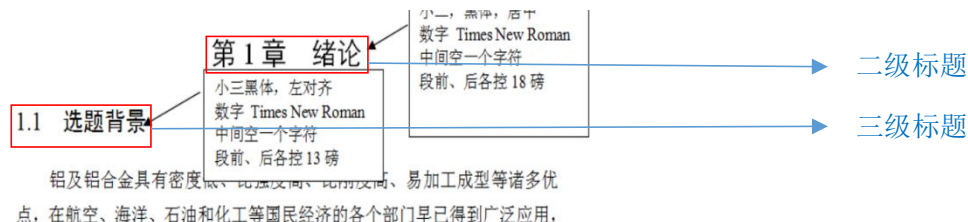


图 2-2-5-1-3：标题等级划分示例

2.2.5.2 标题示例

下面给出了一个包含标题要素的图片示例：

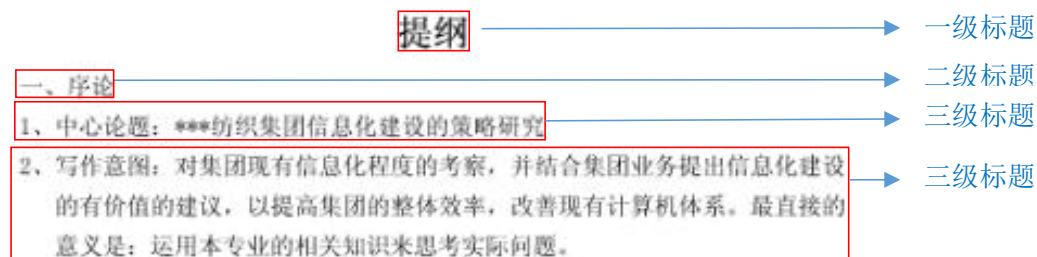


图 2-2-5-2-1：标题示例

上图中，有两个三级标题，最后一个三级标题由三行文字组成。上图对应 JSON 结果如下：

```
{
  "id": "1",
  "type": "layout",
  ...,
  "content":
  [
    [
      //一级标题
      {
        "type": "title",
        ...,
        "level": 1,
        "content":
        [
          [
            {
              "type": "paragraph",
              ...,
              "content":
              [
                [
                  {
                    "type": "textline",
                    ...,
                    "content": [[...]],
                    "text": ["提纲"]
                  }
                ]
              ]
            }
          ]
        ]
      },
      // 二级标题
      {
        "type": "title",
        ...,
        "level": 2,
        "content":
        [
          [
            {
              "type": "paragraph",
              ...,
              "content":
              [
                [

```

```

{
  "type": "textline",
  ...,
  "content": [[...]],
  "text": ["一、序论"]
}
]
}
]
}
]
}
]
...
}
]
}

```

2.2.6 段落

2.2.6.1 结构说明

段落（paragraph）是指语义相对独立的一段文本内容，段落内容由一条或者多条相邻的文本行组成。段落节点字段说明如下：

paragraph			
字段名称	字段类型	是否必须	说明
id	String	是	编号
type	String	是	类型，填写paragraph，表示段落要素
angle	Float	是	v1.1版本新增： 角度，取值范围[0, 360)，按照顺时针方向
coord	Point[]	是	段落的位置，是段落对应区域的最大外接矩形坐标
contour	Point[]	是	段落的轮廓，是段落对应区域的轮廓坐标
attribute	Object[]	是	段落的属性集合
content	Textline[][]	是	子要素集合，子要素仅可包含文本行（textline）要素。其中，第一个维度表示多候选，第二个维度表示子节点集合
text	String[]	是	段落对应的多候选文字内容。 不同文本行之间用\n分隔

段落的属性（attribute）集合字段说明：

属性	字段名	字段类型	字段说明
关联要素	name	String	relation

	value	String[]	存放属于同一段落要素的编号（id）
--	-------	----------	-------------------

下图给出了一个段落示例：

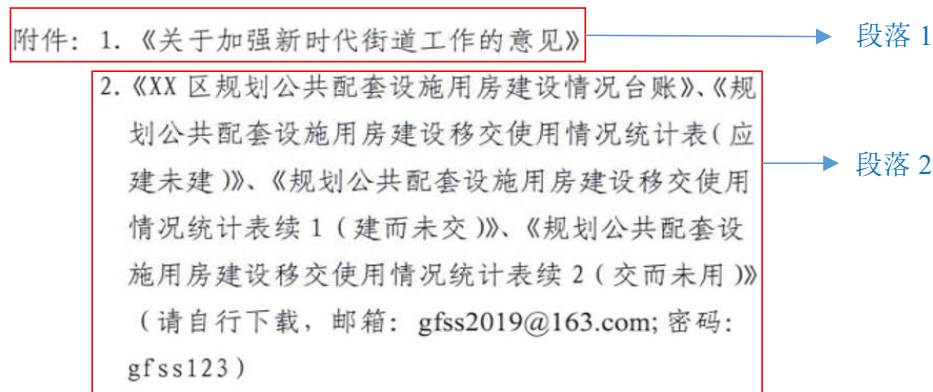


图 2-2-6-1-1：段落示例

上图包含两个段落，当相邻的文本行内容不相关时，它们会被视作独立的段落。

2.2.6.2 段落示例

下图给出了一个段落示例：

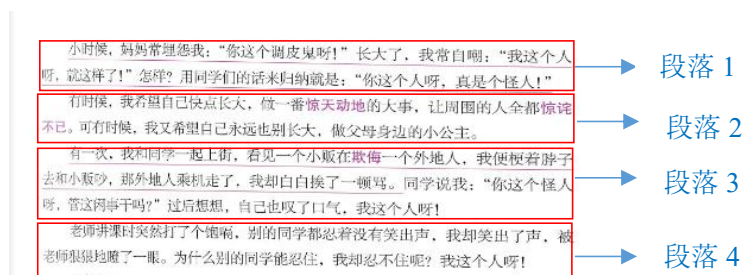


图 2-2-6-2-1：段落示例

上图中，版面内包含四个段落，上图对应 JSON 结果如下：

```
{
  "type": "layout",
  "...",
  "content": [
    [
      // 段落 1
      {
        "id": "1",
        "type": "paragraph",
        "coord": [],
        "contour": [],
        "attribute": [],
        "content": [
          [
            {
              "type": "textline",
              ...
            }
          ]
        ]
      }
    ]
  ]
}
```

```
}  
  ]  
  ],  
  "text":["小时候，妈妈常埋怨我..."]  
},  
...  
  ]  
}
```

2.2.7 文本行

2.2.7.1 结构说明

文本行（**textline**）是指图片中文字相对独立的一行文本对应的区域，文本行内只有一行文字。文本行由文本块要素（**text_block**）组成，文本行子要素仅可包含文本块要素，并且文本行要素仅可作为子要素存在于段落（**paragraph**）节点下。文本行节点字段说明如下：

textline			
字段名称	字段类型	是否必须	说明
id	String	是	编号
type	String	是	类型，填写textline，表示文本行要素
angle	Float	是	文本行的文字角度，取值范围[0, 360)，按照顺时针方向
direction	String	是	文字朝向，取值有： vertical ：竖排文字 horizontal ：横排文字
score	Float	是	得分，取值范围[0, 1]
coord	Point[]	是	文本行的位置，是文本行对应区域的最大外接矩形坐标
contour	Point[]	是	文本行的轮廓，是文本行对应区域的轮廓坐标
attribute	Object[]	是	文本行的属性集合，可以设置文本行缩进、对齐、阅读顺序
content	Text_block[][]	是	子要素集合，仅可包含文本块（ text_block ）。其中，数组的第一个维度表示多候选，第二个维度表示子节点集合。
text	String[]	是	文本行对应的多候选文字内容。

下图给出一个文本行示例：



图 2-2-7-1-1: 文本行示例

当同一行上的文字之间有多个空格, 但文字之间有明显的语义关系时, 也当作同一文本行, 下图给出一个示例:

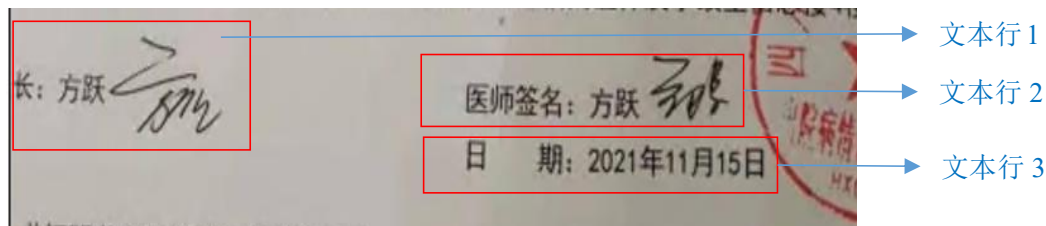


图 2-2-7-1-2: 文本行示例

上图中, “日期”之间有多个空格, 因为日期是一个词语, 所以仍当作一个文本行要素输出, 图中还包含印章要素, 为了便于说明, 未在图中画出。

2.2.7.2 文本行朝向

朝向 (direction) 用来标记文本行是否是竖排文字, 这里的竖排文字要求文字是自上而下书写的, 横向文本旋转得到的文字不属于竖排文字。例如横向文字旋转 90 度或者 270 度得到文字不是竖排文字。

类别简称	类别名称
vertical	竖排文字
horizontal	横排文字

下图给出了一个文本行朝向的示例:

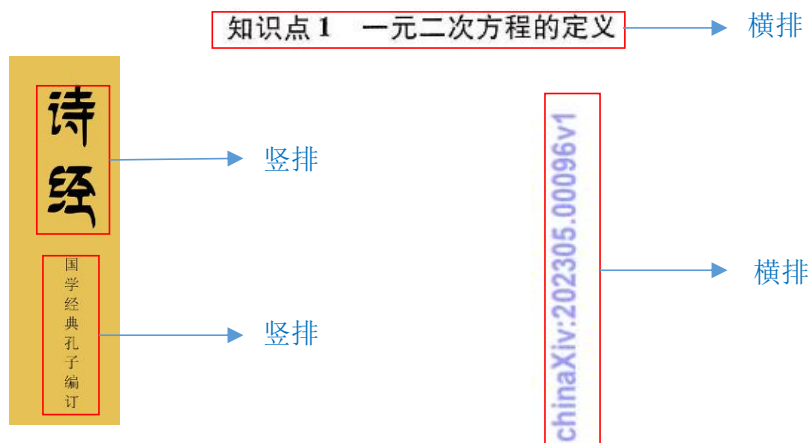


图 2-2-7-2-1：横向文本行示例

2.2.7.3 文本行属性

文本行具有缩进、对齐、阅读顺序等属性，这些属性存放在属性 `attribute` 节点下，具体见下表：

属性	字段名	字段类型	字段说明
缩进字符数	name	String	indent
	value	Int	文本行缩进字符个数，用数字表示
对齐方式	name	String	alignment
	value	String	对齐方式，取值有： left: 左对齐 center: 居中 right: 右对齐
阅读顺序	name	String	reading_order
	value	String	阅读顺序，取值有： left_right: 从左到右 right_left: 从右到左 top_down: 从上到下 down_top: 从下到上

下图给出了一个文本行示例：

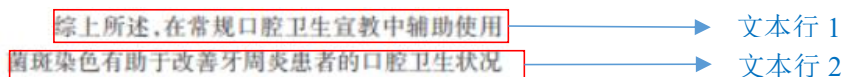


图 2-2-7-3-1：文本行属性示例

上图中，文本行 1 缩进 2 个字符，左对齐，从左到右的阅读顺序；文本行 2 无缩进，左对齐，从左到右的阅读顺序。

2.2.7.4 文本行示例

下图给出了一个文本行示例：

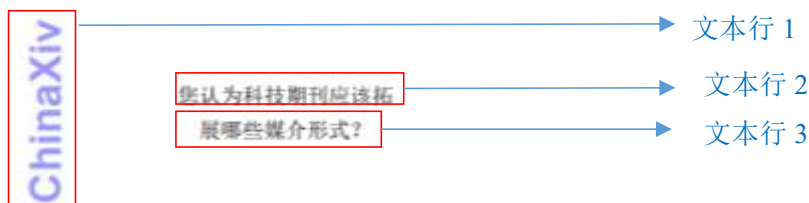


图 2-2-7-4-1：文本行示例

上图中，文本行 1、文本行 2、文本行 3 均为非竖排文字，其中文本行 1 阅读顺序是从下到上（down_top），文本行 3 的阅读顺位是从左到右（left_right）。

上图对应的 JSON 结果如下：

```
{
  "type": "paragraph",
  "content": [
    [
      // 文本行 1
      {
        "type": "textline",
        "direction": "horizontal",
        "attribute": [
          {
            "name": "indent",
            "value": "0"
          },
          {
            "name": "alignment",
            "value": "center"
          },
          {
            "name": "reading_order",
            "value": "down_top"
          }
        ]
      },
      "content": [...],
      "text": ["Chinaxiv"]
    ],
    // 文本行 2
    {
      "type": "textline",
      "direction": "horizontal",
      "attribute": [
        {
          "name": "indent",
          "value": 0
        },
        {
          "name": "alignment",
          "value": "left"
        },
        {
          "name": "reading_order",
          "value": "left_right"
        }
      ]
    },
    "content": [...],
    "text": [
      "您认为科技期刊应该扩",
      "你认为科技期刊应该广"
    ]
  ],
  ...
]
```

2.2.7.5 文本块

2.2.7.5.1 结构说明

文本块（text_block）是指文本行中一段逻辑属性相同的连续文本，这里的逻辑属性是指文本的内容，例如如普通文本、公式、作答区等。文本块节点的子要素仅可包含文本块（text_block）或文本单元（text_unit）。文本块节点的字段说明如下：

text_block			
字段名称	字段类型	是否必须	说明
id	String	是	编号
type	String	是	text_block，表示文本块节点
coord	Point[]	是	文本块的位置，是文本块对应区域的最大外接矩形坐标
contour	Point[]	是	文本块的轮廓，是文本块对应区域的轮廓坐标
category	String	是	文本块类型，取值有： text：纯文本 formula：行内公式 answer：行内作答区 question_number：题号 item_number：列表项序号、修饰符 graph：行内插图
attribute	Object[]	是	文本块的属性集合
content	Object[][]	是	子要素集合，子要素仅可包含文本块（text_block）、文本单元（text_unit）。其中，数组的第一个维度表示多候选，第二个维度表示子节点集合。
text	String[]	是	文本块对应的多候选文字内容

下图给出了一个文本块示例：

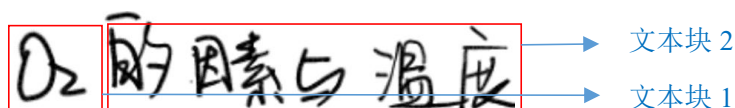


图 2-2-7-5-1-1：中文公式混排场景文本块划分

上图中，文本行包含公式部分和中文部分，其中公式部分作为一个文本块，中文部分作为另一个文本块。

2.2.7.5.2 文本块类别

文本块有多种类别，以下是具体说明：

字段名称	字段说明
text	文本
formula	公式，LaTeX格式
graph	行内插图
item_number	列表项序号、修饰符

下图给出了一个类别为行内插图的文本块示例：

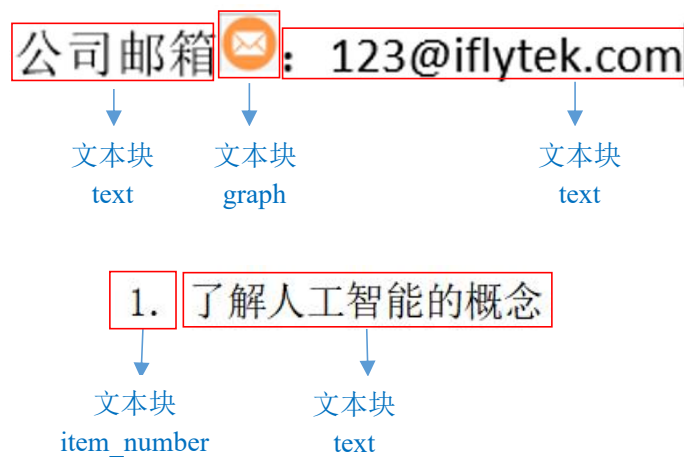


图 2-2-7-5-2-1：文本块示例

2.2.7.5.3 文本块属性

文本块具有关联属性（relation），关联属性表示一个文本块横跨两个文本行这种情况，文本块属性具体说明如下：

属性	字段名	字段类型	字段说明
关联要素	name	String	relation
	value	String[]	存放属于同一文本块要素的编号（id）

2.2.7.5.4 文本块示例

下图给出了一个包含文本块要素的示例：

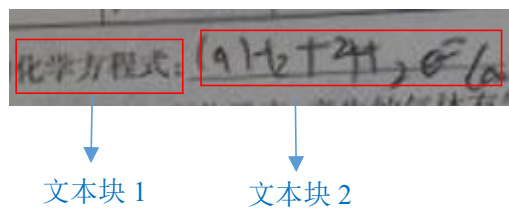


图 2-2-7-5-4-1：文本块示例

上图中，文本行要素包含 2 个文本块，文本块 1 的类别为文本（text），文本块 2 的类别为公式（formula），文本块内包含文本单元要素，为了便于说明，上图中仅标注文本块要素。上图对应 JSON 结果如下：

```
{
  "type": "textline",
  "attribute": [],
  ...
  "content":
  [
    [
      // 文本块 1 text
      {
        "type": "text_block",
        "category": "text",
        ...
        "content": [[...]],
        "text": ["化学方程式: "]
      },
      // 文本块 2 formula
      {
        "type": "text_block",
        "category": "formula",
        ...
        "content": [[...]],
        "text": ["C^{a}H^{2}..."]
      }
    ]
  ]
}
```

2.2.7.6 文本单元

2.2.7.6.1 结构说明

文本单元（text_unit）是文本块中一段属性相同的连续文本，这里的属性指的是文本的外观属性（如字体、字号、倾斜、加粗等）或内在属性（错别字、模糊等）。不同属性的连续文本作为单独的文本单元输出。文本单元仅可作为子要素出现在文本块（text_block）内，文本单元节点字段说明如下：

text_unit			
字段名称	字段类型	是否必须	说明
id	String	是	编号
category	String	是	文字单元类别，取值有： handwrite: 手写体 print: 印刷体
type	String	是	text_unit，表示文本单元节点
attribute	Object[]	是	文本单元的属性集合
text	String	是	文本单元字符串
word	Word_unit[][]	否	词集合，中文是单个字，英文是单词。其中，数组第一个维度表示文本单元中包含的所有字或单词；第二个维度表示字或单词的多候选
char	Char_unit[][]	否	字符集合，中文是单个字，英文是单个字母。其中，数组的第一维度表示文本单元中包含的所有字符；第二个维度表示字符的多候选

2.2.7.6.2 文本单元属性

文本单元属性分为外观属性和内在属性，外观属性是文本的外在特征，例如前景色、字号等，内在属性是文本的语义特征，例如模糊、完全涂抹等。只有文本单元包含某个属性时属性字段才会给出字段名，若无属性则为输出结果为空的数组。文本单元的外观属性说明如下：

外观属性	字段名	字段类型	字段说明
加粗	name	String	bold
斜体	name	String	italic
背景色	name	String	background_color
	value	String	颜色值，颜色RGB值十六进制字符串，示例： #FF0000: 红色 #00FF00: 绿色 #0000FF: 蓝色 #FFFF00: 黄色 #800080: 紫色 #FFA500: 橙色 #00FFFF: 青色 #FFFFFF: 白色

			#A52A2A: 棕色 #AAAAAA: 灰色 #FFC0CB: 粉色 #000000: 黑色
前景色	name	String	foreground_color
	value	String	颜色值，颜色RGB值十六进制字符串，取值同背景色
字号	name	String	font_size按照像素数输出
	value	Int	像素单位（pixel）
下划线	name	String	underline
上划线	name	String	overline
中间删除线	name	String	strikethrough

外观属性中的背景色属性（background_color）是指文本单元的背景颜色；前景色属性（foreground_color）是指文本单元的字体颜色；字号属性（font_size）大小用像素表示；加粗（bold）、斜体（italic）属性只有属性名称 name 字段，没有 value 字段。下图给出了一个包含不同外观属性文本单元的图片示例：

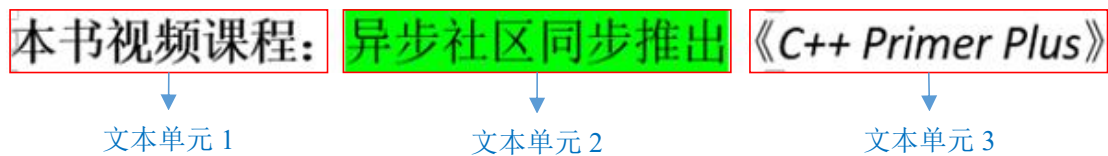


图 2-2-7-6-2-1：文本单元外观属性示例

上图中，文本单元 1 具有加粗（bold）外观属性，文本单元 2 背景色（background_color）属性的 value 值为绿色（#008000），文本单元 3 具有斜体（italic）属性，上图对应 JSON 结果如下：

```
{
  ...
  [
    // 文本单元 1 加粗属性
    {
      "type": "text_unit",
      "attribute": [
        {
          "name": "bold"
        },
        ...
      ],
      "text": "本书视频课程"
    },
    // 文本单元 2 绿色背景色属性
    {
      "type": "text_unit",
      "attribute": [
        {
          "name": "background_color",
          "value": "#008000"
        },
        ...
      ],
      "text": "异步社区同步推出"
    },
    {
      "type": "text_unit",
      "attribute": [
        {
          "name": "italic"
        },
        ...
      ],
      "text": "《C++ Primer Plus》"
    }
  ]
}
```

```
    },
    "text": "异步社区同步推出"
  },
  // 文本单元 3 斜体属性
  {
    "type": "text_unit",
    "attribute": [
      {
        "name": "itliac"
      },
      ...
    ],
    "text": "<<c++ Primer Plus>>"
  },
]
```

文本单元的内在属性说明如下：

内在属性	字段名	字段类型	字段说明
模糊	name	String	ambiguous,
候选词	name	String	candidate
	value	String[]	候选词列表
完全涂抹	name	String	smear_full, 如果文本单元具有该属性, 则文本单元的text为空字符串
	value	String	\smear构成的字符串, 有几个完全涂抹字符给几个\smear字符
缩进	name	String	indent
	value	Int	缩进个数, 数字表示

下面结合示例说明各属性如何使用。

示例 1：模糊和候选词属性示例

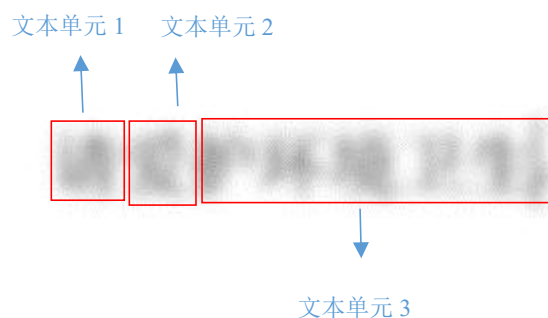


图 2-2-7-6-2-2：模糊和候选词属性示例

上图中，当图片内容比较模糊，文本单元识别结果不确定时，使用模糊（ambiguous）属性表示，例如上图中的文本单元 1、文本单元 2 和文本单元 3 都具有模糊属性。此外，部分文本单元还存在文本内容有多候选情况，使用候选

词（candidate）属性表示，例如上图中的文本单元 2。上图对应的 JSON 结果如下：

```
{
  // 文本单元 1
  {
    "type": "text_unit",
    "...",
    "attribute": [
      {
        "name": "ambiguous"
      }
    ],
    "text": "请"
  },
  // 文本单元 2
  {
    "type": "text_unit",
    "...",
    "attribute": [
      {
        "name": "ambiguous"
      },
      {
        "name": "candidate",
        "value": ["保"]
      }
    ],
    "text": "爱"
  },
  // 文本单元 3
  {
    "type": "text_unit",
    "...",
    "attribute": [
      {
        "name": "ambiguous"
      }
    ],
    "text": "保护环境 1"
  }
}
```

示例 2：完全涂抹属性示例

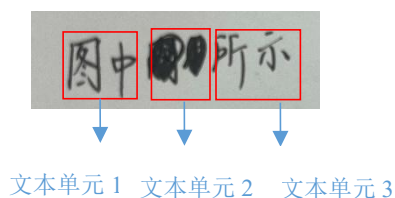


图 2-2-7-6-2-3：完全涂抹属性示例

上图中，文本单元 2 的文字内容完全涂抹不可识别，所以赋予文本单元 2 完全涂抹（smear_full）属性，上图对应的 JSON 结果如下：

```
{
  // 文本单元 1
```

```
{
  "type": "text_unit",
  "category": "handwrite",
  "attribute": [...],
  ...,
  "text": "图中"
},
// 文本单元 2
{
  "type": "text_unit",
  "category": "handwrite",
  "attribute": [
    {
      "name": "smear_full",
      "value": "\smear\smear"
    }
  ],
  ...,
  "text": ""
},
// 文本单元 3
{
  "type": "text_unit",
  "category": "handwrite",
  "attribute": [...],
  ...,
  "text": "所示"
}
}
```

2.2.7.6.3 文本单元示例

下图给出了一个包含不同类别文本单元的图片示例：

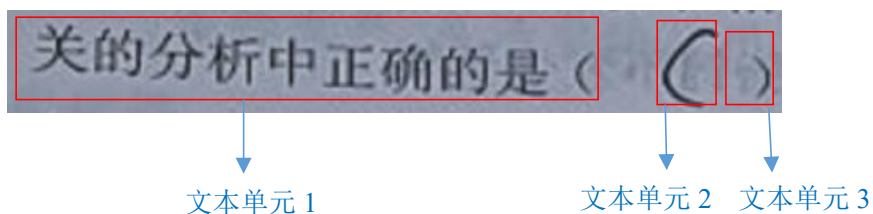


图 2-2-7-6-3-1：文本单元类别示例

上图中包含三个文本单元，其中文本单元 1、文本单元 3 的类别为印刷体（print），文本单元 2 的类别为手写体（handwrite）。上图对应 JSON 结果如下：

```
{
  "type": "textline",
  ...,
  "content": [
    [
      // 文本块 1
      {
        "type": "text_block",
        "category": "text",
        ...,
        "content": [
          [
            // 文本单元 1
```

```
{
  {
    "type": "text_unit",
    "...",
    "category": "print",
    "text": "关的分析..."
  }
]
},
// 文本块 2
{
  "type": "text_block",
  "category": "text",
  "...",
  "content":
  [
    [
      // 文本单元 2
      {
        "type": "text_unit",
        "category": "handwrite",
        "...",
        "text": "C"
      }
    ]
  ]
},
// 文本块 3
{
  "type": "text_block",
  "category": "text",
  "content":
  [
    [
      // 文本单元 3
      {
        "type": "text_unit",
        "category": "print",
        "...",
        "text": ") "
      }
    ]
  ]
}
]
```

2.2.7.7 词单元

2.2.7.7.1 结构说明

词单元（word_unit）仅存在于文本单元的词（word）节点下，汉字、英文单词、拼音、标点等符号是构成词单元的基本单位。词单元节点字段的说明如下：

word_unit			
字段名称	字段类型	是否必须	说明
score	Float	是	得分，取值范围[0, 1]
text	String	是	词识别结果
coord	Point[]	是	词单元的位置，是词单元对应区域的最大外接矩形坐标

2.2.7.8 字符单元

2.2.7.8.1 结构说明

字符单元（char_unit）仅存在于文本单元的字符 char 节点下，汉字、英文字母、拼音字母、标点等符号是构成字符单元的基本单位。字符单元的字段说明如下：

char_unit			
字段名称	字段类型	是否必须	说明
score	Float	是	得分，取值范围[0, 1]
text	String	是	字符识别结果
coord	Point[]	是	字符单元的位置，是字符单元对应区域的最大外接矩形坐标

2.2.7.8.2 词、字符单元示例

下图给出了一个词单元和字符单元的示例：

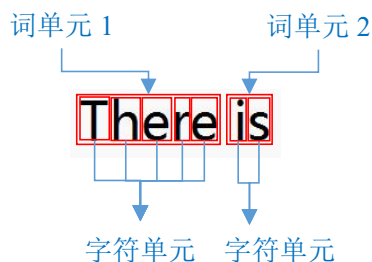


图 2-2-7-8-2-1：词单元和字符单元示例

上图中包含两个词单元和多个字符单元，上图对应的 JSON 结果如下：

```
{
  // 文本单元
  "type": "text_unit",
  "category": "print",
  ...,
  "text": "There is",
  "word":
  [
    // 多候选 top1
    [
      // 词单元
      {
        "score": 0.98,
        "content": "There",
        "coord": []
      },
      {
        "score": 0.90,
        "content": "is",
        "coord": []
      }
    ]
  ],
}
```

```
],  
  // 多候选 top2  
  [  
    {  
      "score": 0.98,  
      "content": "Ther",  
      "coord": []  
    },  
    {  
      "score": 0.98,  
      "content": "Is",  
      "coord": []  
    },  
    ...  
  ],  
],  
"char":  
[  
  // 多候选 top1  
  [  
    // 字符单元  
    {  
      "score": 0.78,  
      "content": "i",  
      "coord": []  
    },  
    ...  
  ],  
  // 多候选 top2  
  [...]  
]
```

2.2.8 表格

2.2.8.1 结构说明

表格（table）是由行和列组成的结构化数据集，可以将不同类型的数据集成在一张表中以显示数据之间存在的某种关系，行通常位于水平方向上，包含某个主题的各种数据内容；列则位于垂直方向上，一般包含某一项数据的详细信息；表格中的每个单元格用于显示某一项具体的数据。

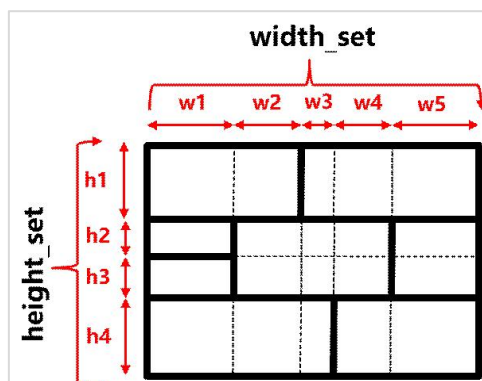


图 2-2-8-1-1：表格示例

单元格是表格的基本组成单位，当表格存在单元格合并的情况时，则以这些单元格的横向、纵向线段为基础，向水平方向和垂直方向延伸，形成一个网格图。该网格图的横向网格数量为 `col`，纵向网格数量为 `row`，每个网格宽度构成的集合为 `width_set`，每个网格高度构成的集合为 `height_set`，其中宽度、高度以像素为单位。

表格一般包含表头，表头用类别为 `head` 的单元格表示，非表头用类别为 `normal` 的单元格表示，在部分业务场景，图片中无表头，根据表格的语义信息可推测出表头，此类表头用类别 `virtual_head` 表示，具体见下一节单元格说明。表题和表注也属于表格内容，存放在 `note` 节点中。表格的字段说明如下：

table			
字段名称	字段类型	是否必须	说明
<code>id</code>	<code>String</code>	是	编号
<code>type</code>	<code>String</code>	是	类型，填写 <code>table</code> ，表示表格要素
<code>angle</code>	<code>Float</code>	是	v1.1版本新增： 角度，取值范围 <code>[0, 360)</code> ，按照顺时针方向
<code>category</code>	<code>String</code>	是	表格类别，类别有： <code>full_line</code> ：有线表 <code>few_line</code> ：少线表 <code>no_line</code> ：无线表
<code>coord</code>	<code>Point[]</code>	是	表格位置，是表格对应区域的最大外接矩形坐标。不包括表题、表注区域
<code>contour</code>	<code>Point[]</code>	是	表格轮廓，是表格对应区域的轮廓，至少包含4个坐标点。不包括表注、表题区域
<code>col</code>	<code>Int</code>	是	表格的列数
<code>row</code>	<code>Int</code>	是	表格的行数
<code>height_set</code>	<code>Int[]</code>	否	表格单元格高度集合，以像素为单位
<code>width_set</code>	<code>Int[]</code>	否	表格单元格宽度集合，以像素为单位
<code>cell</code>	<code>Cell[]</code>	否	单元格集合
<code>note</code>	<code>Region[]</code>	是	表格的说明，存放表格的表题、表注
<code>attribute</code>	<code>Object[]</code>	否	表格属性，见属性字段说明

表格分为有线表、少线标和无线表三类。其中有线表是指表格框线绘制完整的表格；少线表是只为了展示、美观等原因未绘制部分框线的表格；无线表是指未绘制框线的表格。

表格的属性（attribute）字段说明：

属性	字段名	字段类型	字段说明
关联要素	name	String	relation
	value	String[]	存放属于同一表格要素的编号（id）

下图是有线表示例：

项目		最小	典型	最大
电源参数	电压	--	12V	--
	纹波	--	--	50mV
	电流	3A		
电源电流(HDMI 输出, 未接其它外设)	工作电流		250mA	380mA
	待机电流		22mA	30mA
电源电流(LVDS)	3.3V 工作电流		400 mA	1.5A
	5V 工作电流		550 mA	2A
	12V 工作电流		580 mA	2A
电源电流 (V-by-one)	12V 工作电流		580 mA	2A
总输出	电流	--	--	3A
	相对湿度	--	--	80%
环境	工作温度	-10℃	--	60℃
	存储温度	-40℃		80℃

→ 有线表

图 2-2-8-1-2：有线表示例

下图是少线表示例，从图中可以看出，竖向外边框和其内的部分横向和竖向边框线均未绘制。

指 标	2016	2017	2018	2019
发电量（亿千瓦时）	832.32	865.50	856.59	836.70
线损率（%）	6.06	6.03	5.30	2.23
年末发电设备容量（万千瓦）	2 370.65	2 399.67	2 446.88	2 568.29
架空线长度（公里）	9 641.98	9 880.37	10 096.89	10 307.73
1 000KV（公里）	147.34	147.34	147.34	147.34
750KV（公里）	106.14	106.14	106.14	106.14
500KV（公里）	1 149.77	1 149.77	1 151.00	1 151.70
220KV（公里）	3 453.92	3 601.04	3 716.88	3 868.03
110KV（公里）	920.19	993.32	1 028.35	1 159.38
35KV（公里）	3 509.91	3 528.07	3 592.23	3 520.12
电缆长度（公里）	12 645.26	13 626.81	14 323.45	14 944.29
500KV（万千瓦安）	31.26	60.86	60.86	60.42
220KV（万千瓦安）	702.24	721.93	820.07	857.81
110KV（万千瓦安）	1 871.78	2 176.27	2 378.41	2 687.91
35KV（万千瓦安）	10 039.98	10 667.76	11 064.55	11 338.16
公用变电容量（万千瓦安）	16 365.25	17 422.64	18 141.59	18 494.43
500KV（万千瓦安）	5 684.64	6 029.64	6 194.64	6 294.84
220KV-35KV（万千瓦安）	10 680.61	11 393.00	11 946.95	12 199.59
用电最高负荷（万千瓦）	3 138.40	3 268.20	3 094.00	3 131.90

→ 少线表

图 2-2-8-1-3：少线表示例

下图是无线表示例，从图中可以看出，所有边框线均未绘制。

股东名称	股本 (元)	持股比例 (%)
龙高集团	5.968亿	49.19%
华建交通	2.174亿	17.92%
其他 A股公众股东	3.990亿	32.89%
合计	12.132亿	100.00%

无线表

图 2-2-8-1-4：无线表示例

2.2.8.2 表格示例

下图给出了一个有线表示例：

行检测效果统计结果		
数据集名	corr	acc
数据集一		
数据集二		
数据集三		

有线表

图 2-2-8-2-1：表格示例

上图中，有线表包含一个表头（head）类别的单元格和若干个普通单元格（normal），表头类别单元格的文字内容为“行检测效果统计结果”，单元格要素（cell）具体说明见 2.2.8.3 节。上图对应的 JSON 结果示例如下：

```
// 有线表
{
  "type": "table",
  "...",
  "col": 5,
  "row": 5,
  "note": [],
  "category": "full_line",
  "cell":
  [
    // 表头
    {
      "category": "head",
      "...",
      "content": [...]
    },
    // 普通单元格
    {
      "category": "normal",
      "...",
      "content": [...]
    }
  ]
}
```

2.2.8.3 单元格

2.2.8.3.1 结构说明

单元格（cell）是表格的基本组成单位，其中类别 **head** 表示单元格类别为表头，非表头用类别为 **normal** 的单元格表示；在部分业务场景，图片中表格无表头，根据表格的语义信息可推测出表头，此类表头用类别 **virtual_head** 表示，注意此类表头中文字为根据语义推测，不是图片中真实存在的文字。单元格字段说明如下：

cell			
字段名称	字段类型	是否必须	说明
id	String	是	标识符（全局唯一）
type	String	是	类型，填写cell，表示单元格
coord	Point[]	是	单元格的位置，是单元格对应区域的最大外接矩形坐标
contour	Point[]	是	单元格的轮廓，是单元格对应区域的轮廓多边形坐标
category	String	是	类别。取值有： head: 表头 virtual_head: 虚拟表头 normal: 普通单元格
col	Int	是	列号，从1开始计数
row	Int	是	行号，从1开始计数
colspan	Int	是	单元格横跨的列数
rowspan	Int	是	单元格横跨的行数
content	Object[]	否	单元格内元素集合，子要素仅可包含段落（paragraph）、插图（graph）、列表（list）、代码（code）、伪代码（pseudocode）、公式（formula）、键值对（key和value）、目录(contents)、二维码（qrcode）、条形码（barcode）、标题（title）、表格（table）要素

2.2.8.3.2 单元格示例

下图给出了一个 4 行 7 列包含 28 个单元格的表格示例：

单元格 1

月份	欠数单数	生产总单数	欠数单数百分比	欠数个数(PCS)	生产总个数(PCS)	欠数个数百分比
2012 年 06 月	22 单	1082 单	2.03%	-76605 ↓		
2012 年 05 月	10 单	1057 单	0.95%	-59614 ↓		
2012 年 04 月	21 单	1000 单	2.10%	-77074 ↓		

单元格 18

图 2-2-8-3-2-1：单元格示例

上图对应的 JSON 结果如下：

```
{
  // 表格 1
  "type": "table",
  "col": 7,
  "row": 4,
  "category": "full_line",
  "cells":
  [
    // 单元格 1
    {
      "type": "cell",
      "category": "normal",
      "content":
      [
        {
          "type": "paragraph",
          "content":
          [
            [
              {
                "type": "textline",
                "attribute":
                [
                  {
                    "name": "indent",
                    "value": 0
                  },
                  {
                    "name": "alignment",
                    "value": "center"
                  }
                ],
                "content":
                [
                  [
                    {
                      "type": "text_block",
                      "category": "text",
                      "content":
                      [
                        [
                          {
                            "type": "text_unit",
                            "category": "print",
                            "text": "月份",
                          }
                        ]
                      ]
                    }
                  ]
                ]
              }
            ]
          ]
        }
      ]
    }
  ],
  [...]
}
```



```
"type": "cell",
"category": "virtual_head",
...,
"content":
[
  {
    "type": "paragraph",
    ...,
    "content": [[...]],
    "text":["项目"]
  }
],
// 单元格 2
{
  "type": "cell",
  "category": "virtual_head",
  ...,
  "content":
  [
    {
      "type": "paragraph",
      ...,
      "content": [[...]],
      "text":["结果"]
    }
  ]
},
...
// 单元格 16
{
  "type": "cell",
  "category": "normal",
  ...,
  "content":
  [
    {
      "type": "paragraph",
      ...,
      "content": [[...]],
      "text":["%"]
    }
  ]
}
]
```

2.2.9 插图

2.2.9.1 结构说明

插图（graph）用来表示文档页面中的图形、图像信息。插图要素的字段说明如下：

graph			
字段名称	字段类型	是否必须	说明
id	String	是	标识符（全局唯一）
type	String	是	类型，填写graph，表示插图要素
angle	Float	是	v1.1版本新增： 角度，取值范围[0, 360)，按照顺时

			针方向
coord	Point[]	是	插图的位置，是插图对应区域的最大外接矩形坐标；当插图下方存在图注时，coord字段的坐标范围也不包含图注。
contour	Point[]	是	插图的轮廓，插图对应区域的轮廓多边形坐标；当插图下方存在图注时，contour字段的坐标范围也不包含图注。
attribute	Object[]	是	插图的属性集合
content	Object[][]	是	插图子要素集合，子要素仅可包括段落（paragraph）、表格（table）、插图（graph）、键值对（key和value）、目录(contents)、二维码（qrcode）、条形码（barcode）、公式（formula）、标题（title）、列表（list）、代码（code）、伪代码（pseudocode）。其中，数组的第一个维度表示多候选，第二个维度表示子节点集合。
note	Region[]	是	图像的标题或注释

注意：对于文本内出现的图标，不属于插图要素，而是作为文本行的行内插图，通过文本块中类别（category）表示，详见 2.2.7.5.2 节。

插图的属性（attribute）集合字段说明：

属性	字段名	字段类型	字段说明
关联要素	name	String	relation
	value	String[]	存放属于同一插图要素的编号（id）

2.2.9.2 插图示例

下图给出了一个插图示例：

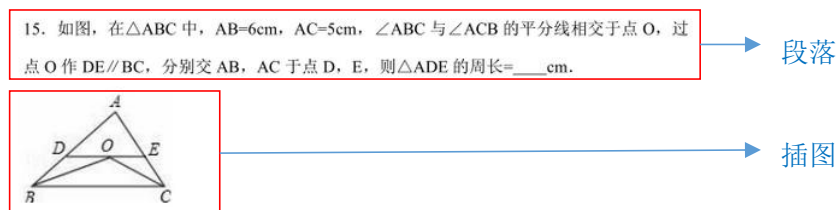


图 2-2-9-2-1：插图

上图中，包含一个段落和一个插图要素，其中插图要素内部有 A、R、C 等文字，图像内的文字均纳入插图子要素中，以段落的形式组织输出，上图对应 JSON 结果如下：

```
{
  "type": "region",
  "category": "column",
  "content": [
    [
      // 插图
      {
        "type": "graph",
        "content": [
          [
            // 图像内文字
            {
              "type": "paragraph",
              "content": [...],
              "text": ["A"]
            },
            {
              "type": "paragraph",
              "content": [...],
              "text": ["R"]
            }
          ],
          [...],
          [...],
        ],
        "note": [...]
      },
      // 段落
      {
        "type": "paragraph",
        "text": ["15. 如图, ..."]
      }
    ]
  ]
}
```

2.2.10 列表

2.2.10.1 结构说明

列表（list）是一种用于表示页面中对齐排列文本的结构，列表有列表项组成。列表的字段说明如下：

list			
字段名称	字段类型	是否必须	说明

id	String	是	标识符（全局唯一）
type	String	是	类型，填写list，表示列表要素
angle	Float	是	v1.1版本新增： 角度，取值范围[0, 360)，按照顺时针方向
coord	Point[]	是	列表的位置，是列表对应区域的最大外接矩形坐标
contour	Point[]	是	列表的轮廓，是列表对应区域的轮廓多边形坐标
attribute	Object[]	是	列表的属性集合
content	Items[][]	是	列表子要素集合，子要素仅可包括列表项（item）要素。其中，数组的第一个维度表示多候选，第二个维度表示子节点集合

列表的属性（attribute）字段说明如下：

属性	字段名	字段类型	字段说明
关联要素	name	String	relation
	value	String[]	存放属于同一列表的编号（id）

2.2.10.2 列表项

列表项（item）是列表的基本组成单位，列表项的具体字段说明如下：

item			
字段名称	字段类型	是否必须	说明
id	String	是	标识符（全局唯一）
type	String	是	类型，填写item，表示列表项
angle	Float	是	v1.1版本新增： 角度，取值范围[0, 360)，按照顺时针方向
coord	Point[]	是	列表项的位置，是列表项对应区域的最大外接矩形坐标
contour	Point[]	是	列表项的轮廓，是列表项对应区域的轮廓多边形坐标
attribute	Object[]	是	列表项的属性集合
content	Object[][]	是	列表项文本内容，仅可为列表（list）、公式（formula）、段落（paragraph）、插图（graph），其中，数组的第一个维度表示多候选，第二个维度表示子节点集合

列表项包含的属性如下：

属性	字段名	字段类型	字段说明
缩进	name	String	indent
	value	Int	缩进字符个数，相对于一行起始位置的缩进字符个数，用数字表示

2.2.10.3 列表示例

下图给出了一个包含列表要素的图片示例：

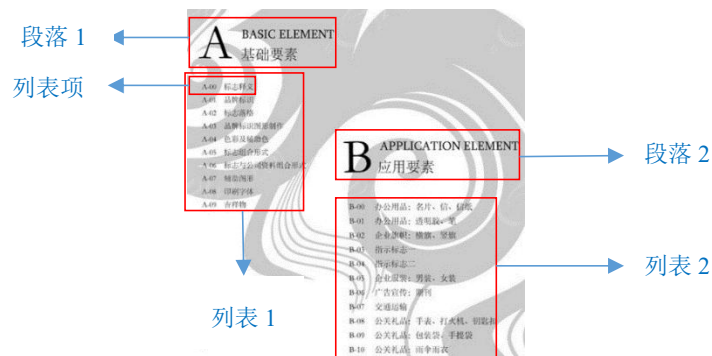


图 2-2-10-3-1：列表与列表项

上图中包含两个列表和两个段落要素，对应的 JSON 结果如下：

```
{
  // 分栏 1
  "type": "region",
  "...",
  "category": "column",
  "content": [
    [
      // 列表 1
      {
        "type": "list",
        "...",
        "indent": 2,
        "content": [
          // top1
          [
            // 列表项
            {
              "type": "item",
              "indent": 0,
              "...",
              "content": [...],
            },
            {
              "type": "item",
              "indent": 0,
              "...",
              "content": [...],
            }
          ],
          [...],
          [...],
        ]
      }
    ]
  ]
}
```

}
]
]

2.2.11 公式

2.2.11.1 结构说明

公式（formula）是指用特定的符号和语言表示的数理化等科学领域中的计算关系。公式分为两种，一种是归属于文本行的行内公式，使用文本块中类别（category）表示，具体说明见 2.2.7.5.2 节，另一种是此节介绍的公式要素，即行间公式。常见公式包括数学公式、物理公式、化学公式等，在此协议中将其分为两类：有机分子式和其他公式。其中有机分子式不可结构化，仅输出其坐标和其中段落内容，其他公式使用 latex 表示，如“ $\frac{\alpha}{2}$ ”表示为“ $\frac{\alpha}{2}$ ”。

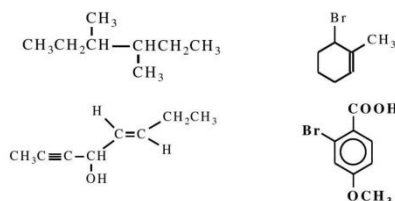


图2-2-11-1-1：有机分子式示例

$$\begin{aligned}\cos^2 \alpha &= \frac{1 + \cos 2\alpha}{2} \\ \sin^2 \alpha &= \frac{1 - \cos 2\alpha}{2}\end{aligned}$$
$$a = \frac{v^2}{r} = \omega^2 r$$
$$\text{Zn} + 2\text{HCl} \rightleftharpoons \text{ZnCl}_2 + \text{H}_2 \uparrow$$
$$(2) \begin{cases} 3x - 5y = 7 \\ 4x + 2y = 5 \end{cases}$$

图2-2-11-1-2：其他公式示例

公式节点的字段说明如下：

formula			
字段名称	字段类型	是否必须	说明
id	String	是	编号
type	String	是	类型，填写formula，表示公式
category	String	否	v1.5版本新增： 公式细分类型，默认值为normal，取值有：

			organic_chemistry:有机分子式 normal:其他类型公式
angle	Float	是	v1.1版本新增: 角度, 取值范围[0, 360), 按照顺时针方向
coord	Point[]	是	公式的位置, 是公式对应区域的最大外接矩形坐标, 不包含注释区域
contour	Point[]	是	公式的轮廓, 是公式对应区域的轮廓多边形坐标, 不包含注释区域
content	Paragraph[][]	是	公式子要素集合, 子要素仅可包括段落(paragraph)要素。其中, 数组的第一个维度表示多候选, 第二个维度表示子节点集合
note	Region[]	否	行间公式的注释文本, 子要素仅可包含区域(Region)要素

2.2.11.2 公式示例

下图给出了一个公式示例:

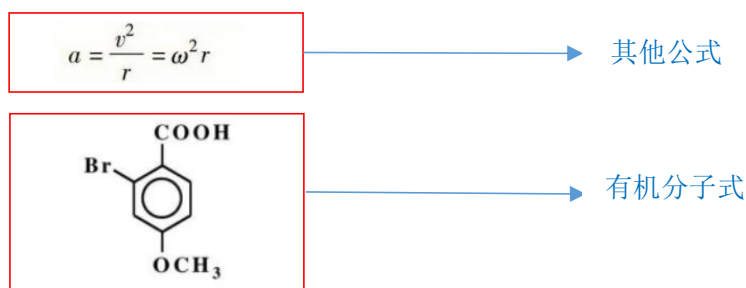


图 2-2-11-2-1: 公式示例

上图对应的 JSON 结果如下:

```
{
  "type": "region",
  "content": [
    [
      // 公式 1: 其他公式
      {
        "type": "formula",
        "category": "normal",
        "content": [
          [
            {
              "type": "paragraph",
              "text": ["a=\frac{v^{2}}{r}=\omega^{2}r"]
            }
          ]
        ]
      },
      // 注释
    ]
  ]
}
```

```
    "note": []
  }
  // 公式 2: 有机分子式
  {
    "type": "formula",
    "category": "organic_chemistry",
    "content":
    [
      [
        {
          "type": "paragraph",
          "text": ["COOH"]
        },
        {
          "type": "paragraph",
          "text": ["Br"]
        },
        {
          "type": "paragraph",
          "text": ["OCH_{3}"]
        }
      ]
    ],
    // 注释
    "note": []
  }
]
```

2.2.12 代码

2.2.12.1 结构说明

代码（code）表示某种编程语言编写的文字，与书写的英文文字的区别在于代码格式固定且遵循一定的排版规范，例如 python 语言要求文字以固定的缩进进行排版。代码的字段说明如下：

code			
字段名称	字段类型	是否必须	说明
id	String	是	编号
type	String	是	类型，填写code，表示代码要素
angle	Float	是	v1.1版本新增： 角度，取值范围[0, 360)，按照顺时针方向
coord	Point[]	是	代码的位置，是代码对应区域的最大外接矩形坐标
contour	Point[]	是	代码的轮廓，是代码对应区域的轮廓多边形坐标

attribute	Object[]	是	代码要素的属性集合
content	Paragraph[][]	是	代码子要素集合，子要素仅可包括段落（paragraph）要素。其中数组的第一个维度表示多候选，第二个维度表示子节点集合，且第二个维度数组个数可为多个。
language	String	是	代码语言类型，类型有： python: python语言 c/c++: c/c++语言 java: java语言 c#: csharp语言 go: GO语言

代码的属性（attribute）字段说明：

属性	字段名	字段类型	字段说明
关联要素	name	String	relation
	value	String[]	存放属于同一代码要素的编号（id）

下图给出一个包含代码要素的示例：

```
int onnxinfer_agent::get_output(unsigned int index,
float* data,unsigned int size,
int dev_type,int dev_id) {
int ret = 0;
float* floatarr = NULL;

ort_api->GetTensorMutableData(output_tensors_[index], (void**)&floatarr);
// 此处需要拷贝到GPU上
if (input_mode_ == IOBINDING)
{
    cudaMemcpyAsync(data, floatarr, size * sizeof(float), cudaMemcpyDeviceToDevice, stream_);

    cudaStreamSynchronize(stream_);
    // ort_api->ReleaseValue(output_tensors_[index]);
}
else
{
    cudaStreamSynchronize(stream_);
    memcpy(data, floatarr, size * sizeof(float));
}
return 0;
}
```

代码

图 2-2-12-1-1：代码示例

2.2.12.2 代码示例

下图给出了一个 C++代码示例：



图 2-2-12-2-1：代码示例

代码子要素仅可包含段落（paragraph）要素，具体的代码文字和格式信息均存放于段落要素中，上图代码内包含多个上图对应的 JSON 结果如下：

```
{
  "type": "region",
  "category": "column",
  "...",
  "content": [
    [
      {
        "type": "code",
        "...",
        "content": [
          // 多候选 top1
          [
            // 段落 1
            {
              "type": "paragraph",
              "...",
            },
            "...",
          ],
          // 多候选 top2
          [...]
        ]
      }
    ]
  ]
}
```

2.2.13 伪代码

2.2.13.1 结构说明

伪代码（pseudocode）是一种用于描述算法逻辑的非正式编程语言，该要素仅可作为子要素存在于区域（region）要素中。伪代码字段说明如下：

pseudocode

字段名称	字段类型	是否必须	说明
id	String	是	编号
type	String	是	类型，填写pseudocode，表示伪代码要素
angle	Float	是	v1.1版本新增： 角度，取值范围[0, 360)，按照顺时针方向
coord	Point[]	是	伪代码的位置，是伪代码对应区域的最大外接矩形坐标
contour	Point[]	是	伪代码的轮廓，是伪代码对应区域的轮廓多边形坐标
attribute	Object[]	是	伪代码的属性集合
content	Paragraph[][]	是	伪代码子要素集合，子要素仅可包括段落（paragraph）、公式（formula）要素，且段落子要素的位置和轮廓与代码要素保持一致。其中，数组的第一个维度表示多候选，第二个维度表示子节点集合

伪代码的属性（attribute）字段说明：

属性	字段名	字段类型	字段说明
关联要素	name	String	relation
	value	String[]	存放属于同一伪代码要素的编号（id）

2.2.13.2 伪代码示例

下图给出了一个包含伪代码要素的图片示例：

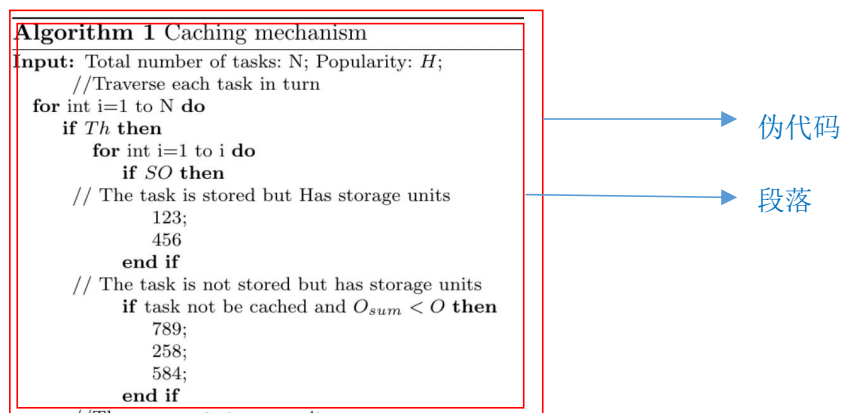


图 2-2-13-2-1：伪代码示例

与代码要素相同，伪代码子要素仅可包含段落（paragraph）要素，具体的代码文字和格式信息均存放于段落要素中，上图对应的 JSON 结果如下：

```
{
  "type": "region",
  "category": "column",
  ...,
  "content":
  [
    [
      {
        "type": "pseudocode",
        ...,
        "content":
        [
          // 多候选 top1
          [
            // 段落
            {
              "type": "paragraph",
              ...
            }
          ],
          // 多候选 top2
          [...]
        ]
      }
    ]
  ]
}
```

2.2.14 信息栏

2.2.14.1 结构说明

信息栏（information_bar）是文档上用于标记装订位置、说明填写规范的区域，一般出现在文档左侧或者右侧，例如装订线。信息栏仅可作为子要素存在页面（page）要素中，信息栏节点字段说明如下：

information_bar			
字段名称	字段类型	是否必须	说明
id	String	是	编号
type	String	是	类型，填写information_bar，表示信息栏要素
angle	Float	是	v1.1版本新增： 角度，取值范围[0, 360)，按照顺时针方向
coord	Point[]	是	信息栏的位置，是信息栏对应区域的最大外接矩形坐标
contour	Point[]	是	信息栏的轮廓，是信息栏对应区域的

			轮廓坐标
content	Object[][]	是	子要素集合，子要素仅可包含标题（title）、段落（paragraph）、表格（table）、列表（list）、伪代码（pseudocode）、代码（code）、脚注（footnote）、公式（formula）、插图（graph）、批注（annotation）、键值对（key和value）、目录(contents)、二维码（qrcode）、条形码（barcode）要素。其中，数组的第一个维度表示多候选，第二个维度表示子节点集合

下图给出一个包含信息栏要素的示例：



图 2-2-14-1-1：信息栏示例

上图包含信息栏、版面、段落等要素，为了便于说明，仅标注出信息栏要素。

2.2.14.2 信息栏示例

下图给出了一个信息栏示例：



图 2-2-14-2-1： 信息栏示例

上图中除了包含信息栏以外，还包含版面、标题、段落等要素，为了便于说明，仅标注出了信息栏要素。上图对应的 JSON 结果如下：

```
{
  "id": "xxx",
  "type": "page",
  "...",
  "content":
  [
    [
      {
        "id": "xxx",
        "type": "information_bar",
        "coord": [...],
        "contour": [...],
        "content":
        [
          // 多候选 top1
          [
            {
              "id": "xxx",
              "type": "paragraph",
              "attribute": [...],
              "text": ["arXiv:2307..."]
            }
          ],
          // 多候选 top2
          [...],
          // 多候选 top3
          [...]
        ]
      }
    ]
  ]
}
```

2.2.15 印章

2.2.15.1 结构说明

印章（seal）是用特定设备在文件上留下的标记，用于表示鉴定或签署，此处的设备有钢印、蜡印、电子签章等。印章仅可作为子要素存在于页面（page）要素中，印章节点字段说明如下：

seal			
字段名称	字段类型	是否必须	说明
id	String	是	编号
type	String	是	类型，填写seal，表示印章要素
angle	Float	是	v1.1版本新增： 角度，取值范围[0, 360)，按照顺时针方向
coord	Point[]	是	印章的位置，是印章对应区域的最大外接矩形坐标
contour	Point[]	是	印章的轮廓，是印章对应区域的轮廓坐标
attribute	Object[]	是	印章的属性集合
content	Object[][]	是	子要素集合，引擎仅输出段落（paragraph）、公式（formula）要素。其中，数组的第一个维度表示多候选，第二个维度表示子节点集合

下图给出了一个包含印章要素的示例：



图 2-2-15-1-1：印章示例

2.2.15.2 印章属性

印章属性字段（attribute）描述了印章的外观特性及类型，详细说明如下：

属性	字段名	字段类型	字段说明
----	-----	------	------

形状	name	String	shape
	value	String	印章的形状，取值有： ellipse: 椭圆形 rectangle: 矩形 circle: 圆形 triangle: 三角形 rhombus: 菱形 other: 其他
颜色	name	String	color
	value	String	印章的颜色值：取RGB值十六进字符或null、other，取值有： #FF0000: 红色 #00FF00: 绿色 #0000FF: 蓝色 #FFFF00: 黄色 #800080: 紫色 #FFA500: 橙色 #00FFFF: 青色 #FFFFFF: 白色 #000000: 黑色 transparent: 透明，一般指钢印 other: 其他颜色（有颜色，但目前不支持）
类型	name	String	type
	value	String	印章的类型,取值有： official: 公章 personal: 个人章 special: 专用章 contract: 合同专用章 finacial: 财务专用章 invoice: 发票专用章 business: 业务专用章
骑缝	name	String	across_page
不完整	name	String	incomplete

2.2.15.3 印章示例

下图给出的示例包含两个红色印章，其中印章 2 内有五角星，协议规定印章内容（content）字段仅输出段落（paragraph）要素，印章 2 内包含两个段落。



图 2-2-15-3-1 签章示例

上图还包括段落、表格等要素，为了便于说明，图中仅标出印章要素。上图对应 JSON 结果如下：

```
{
  "id": "xxx",
  "type": "page",
  "category": "document",
  ...
  "content":
  [
    [
      // 印章 1
      {
        "id": "xxx",
        "type": "seal",
        ...
        "attribute":
        [
          {
            "name": "shape",
            "value": "circle"
          },
          {
            "name": "color",
            "value": "#FF0000"
          },
          {
            "name": "type",
            "value": "official"    // 公章
          }
        ],
        "content":
        [
          // 多候选 top1
          [
            {
              "id": "xxx",
              "type": "paragraph",
              "attribute": [...],
              ...
              "text": [...]
            }
          ]
        ]
      },
      // 印章 2
      {
        "id": "xxx",
        "type": "seal",
        ...
        "attribute":
        [
          {
            "name": "shape",
            "value": "circle"
          },
          {
            "name": "color",
            "value": "#FF0000"
          },
          {
            "name": "type",
            "value": "official"    // 公章
          }
        ],
        "content":
        [
          // 多候选 top1
          [
            {
              "id": "xxx",
              "type": "paragraph",
              "attribute": [...],
              ...
              "text": [...]
            }
          ]
        ]
      }
    ]
  ],
  ...
}
```

```
{
  {
    "name": "color",
    "value": "#FF0000"
  },
  {
    "name": "type",
    "value": "official"    // 公章
  }
],
"content":
[
  [
    {
      "id": "xxx",
      "type": "paragraph",
      "attribute": [...],
      "text": [...]
    },
    {
      "id": "xxx",
      "type": "paragraph",
      "attribute": [...],
      "text": [...]
    }
  ]
]
}
```

2.2.14.1 结构说明

信息栏（`information_bar`）是文档上用于标记装订位置、说明填写规范的区域，一般出现在文档左侧或者右侧，例如装订线。信息栏仅可作为子要素存在页面（`page`）要素中，信息栏节点字段说明如下：

information_bar			
字段名称	字段类型	是否必须	说明
id	String	是	编号
type	String	是	类型，填写 <code>information_bar</code> ，表示信息栏要素
angle	Float	是	v1.1版本新增： 角度，取值范围 <code>[0, 360)</code> ，按照顺时针方向
coord	Point[]	是	信息栏的位置，是信息栏对应区域的最大外接矩形坐标
contour	Point[]	是	信息栏的轮廓，是信息栏对应区域的轮廓坐标
content	Object[][]	是	子要素集合，子要素仅可包含段落（ <code>paragraph</code> ）、表格（ <code>table</code> ）、插图（ <code>graph</code> ）要素。其中，数组的第一个

		维度表示多候选，第二个维度表示子节点集合
--	--	----------------------

下图给出一个包含信息栏要素的示例：

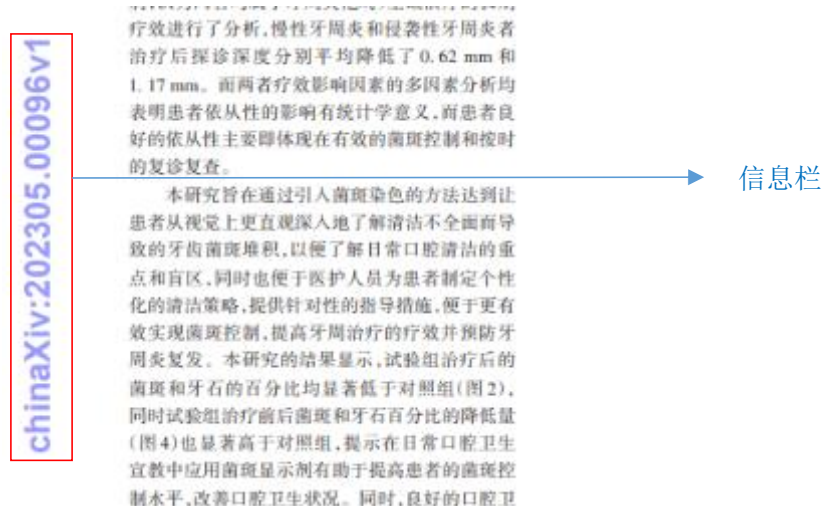


图 2-2-14-1-1：信息栏示例

上图包含信息栏、版面、段落等要素，为了便于说明，仅标注出信息栏要素。

2.2.14.2 信息栏示例

下图给出了一个信息栏示例：



图 2-2-14-2-1：信息栏示例

上图中除了包含信息栏以外，还包含版面、标题、段落等要素，为了便于说明，仅标注出了信息栏要素。上图对应的 JSON 结果如下：

```
{
  "id": "xxx",
  "type": "page",
  "...",
  "content":
  [
    [
      {
        "id": "xxx",
        "type": "information_bar",
        "coord": [...],
        "contour": [...],
        "content":
        [
          // 多候选 top1
          [
            {
              "id": "xxx",
              "type": "paragraph",
              "attribute": [...],
              "text": ["arXiv:2307..."]
            }
          ],
          // 多候选 top2
          [...],
          // 多候选 top3
          [...]
        ]
      }
    ]
  ]
}
```

2.2.16 指印

2.2.16.1 结构说明

指印（fingerprint）是用于确认个人信息按压在文档上的捺印，不包含指印下的文字。指印仅可作为子要素存在页面（page）要素中，无 content 节点。指印的字段说明如下：

fingerprint			
字段名称	字段类型	是否必须	说明
id	String	是	编号
type	String	是	类型，填写fingerprint，表示指印要素
angle	Float	是	v1.1版本新增： 角度，取值范围[0, 360)，按照顺时针方向

coord	Point[]	是	指印的位置，是指印对应区域的最大外接矩形坐标
contour	Point[]	是	指印的轮廓，是指印对应区域的轮廓坐标
attribute	Object[]	是	指印的属性集合

指印要素如下图所示：

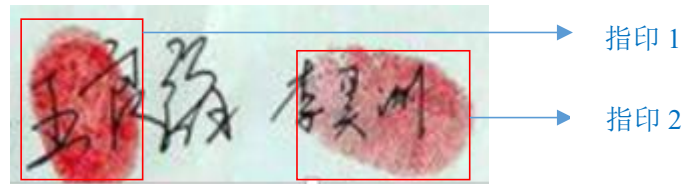


图 2-2-16-1-1：指印示例

2.2.16.2 指印属性

指印属性指的是指印的外观属性，目前仅支持背景颜色：

属性	字段名	字段类型	字段说明
指印颜色	name	String	background_color
	value	String	指印的颜色值：取RGB值的十六进制字符，取值有： #FF0000：红色 #00FF00：绿色 #0000FF：蓝色 #FFFF00：黄色 #800080：紫色 #FFA500：橙色 #00FFFF：青色 #FFFFFF：白色 #000000：黑色

2.2.16.3 指印示例

下图给出了一个指印示例，该示例中有三个红色指印。

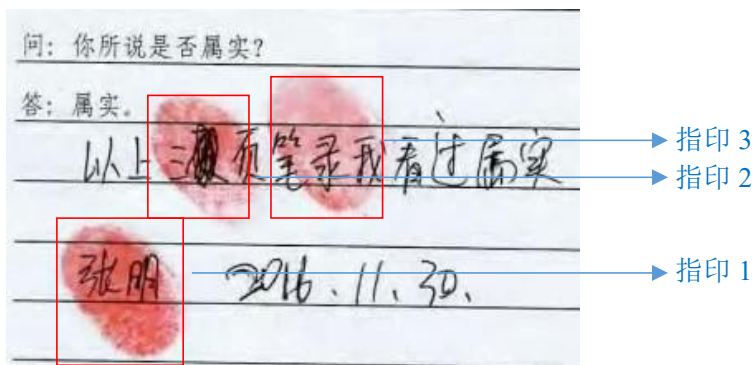


图 2-2-16-3-1: 指印示例

上图中除了包含指印以外, 还包括段落、文本行等要素, 为了便于说明, 仅标注了指印要素, 上图对应 JSON 结果如下:

```
{
  "id": "xxx",
  "type": "page",
  "...",
  "content":
  [
    [
      // 指印 1
      {
        "id": "1",
        "type": "fingerprint",
        "coord": [...],
        "contour": [...],
        "attribute":
        [
          {
            "name": "color",
            "value": "#FF0000"
          }
        ]
      },
      // 指印 2
      {
        "id": "2",
        "type": "fingerprint",
        "coord": [...],
        "contour": [...],
        "attribute":
        [
          {
            "name": "color",
            "value": "#FF0000"
          }
        ]
      },
      // 指印 3
      {...}
    ]
  ]
}
```

2.2.17 条形码

2.2.17.1 结构说明

条形码（barcode）是由宽度不等的多个黑条和空白交替排列构成的一个图形标识符。条形码仅可作为子要素存在于页面（page）要素下，条形码节点字段说明如下：

barcode			
字段名称	字段类型	是否必须	说明
id	String	是	编号
type	String	是	类型，填写barcode，表示条形码要素
angle	Float	是	v1.1版本新增： 角度，取值范围[0, 360)，按照顺时针方向
coord	Point[]	是	条形码的位置，是条形码对应区域的最大外接矩形坐标。条形码区域包含上下文文字行
contour	Point[]	是	条形码的轮廓，是条形码对应区域的轮廓坐标。条形码区域包含上下文文字行
attribute	Object[]	是	条形码的属性集合
content	Object[][]	是	子要素集合，子要素仅可包含段落（paragraph）、插图（graph）要素。其中，数组的第一个维度表示多候选，第二个维度表示子节点集合

下图给出了一个条形码示例：



图 2-2-17-1-1：条形码

上图中除包含条形码以外，还包含段落、文本行等要素，为了便于说明，图中仅标注出条形码要素。

2.2.17.2 条形码属性

条形码属性节点（attribute）保存了条形码解码后的文本内容。

属性	字段名	字段类型	字段说明
解码文本	name	String	decoded_text
	value	String	条形码解析字符

2.2.17.3 条形码示例

下图给出了一个条形码的示例：



图 2-2-17-3-1：条形码示例

上图中除包含条形码以外，还包含段落、文本行等要素，为了便于说明，图中仅标注了条形码要素，上图对应 JSON 结果如下：

```
{
  "id": "xxx",
  "type": "page",
  "content":
  [
    [
      {
        "id": "xxx",
        "type": "barcode",
        "coord": [...],
        "contour": [...],
        "attribute":
        [
          {
            "name": "decoded_text",
            "value": "..."
          }
        ],
        "content": []
      }
    ]
  ]
}
```

2.2.18 二维码

2.2.18.1 结构说明

二维码（qrcode）是文档上用某种特定的几何图形按一定规律在平面分布的黑白相间的图形符号。二维码仅可作为子要素存在页面（page）要素中，二维码节点字段说明如下：

qrcode			
字段名称	字段类型	是否必须	说明
id	String	是	编号
type	String	是	类型，填写qrcode，表示二维码要素
angle	Float	是	v1.1版本新增： 角度，取值范围[0, 360)，按照顺时针方向
coord	Point[]	是	二维码的位置，是二维码对应区域的最大外接矩形坐标
contour	Point[]	是	二维码的轮廓，是二维码行对应区域的轮廓坐标
attribute	Object[]	是	二维码的属性集合
content	Object[][]	是	子要素集合，子要素仅可包含插图（graph）、段落（paragraph）要素。其中，数组的第一个维度表示多候选，第二个维度表示子节点集合；

下面给出了一个包含二维码要素的图片示例：

新能源汽车商业保险保险单



二维码

图 2-2-18-1-1：二维码示例

2.2.18.2 二维码属性

二维码属性节点（attribute）描述了二维码的形状及文本内容，二维码属性说明如下：

属性	字段名	字段类型	字段说明
形状	name	String	shape

	value	String	二维码形状: circle: 圆形 rectangle: 方形
解码文本	name	String	decoded_text
	value	String	解码后的文字内容

2.2.18.3 二维码示例

下图给出了一个二维码示例，



图 2-2-18-3-1: 二维码示例

上图中除了包含二维码要素以外，还包含段落、插图等要素，为了便于说明，图中仅标注了二维码要素，上图对应的 JSON 结果如下：

```
{
  "id": "1",
  "type": "page",
  ...,
  "content":
  [
    [
      {
        "id": "2",
        "type": "qrcode",
        "coord": [...],
        "contour": [...],
        "attribute":
        [
          {
            "name": "shape",
            "value": "rectangle"
          },
          {
            "name": "decoded_text",
            "value": "..."
          }
        ]
      },
      ...
    ],
    "content":
    [
      ...
    ]
  ]
}
```

```
{
  {
    "id": "4",
    "type": "graph",
    "coord": [...],
    "contour": [...],
    "content": [[...]],
    "note": [...]
  },
  [...],
  [...]
}
]
```

2.2.19 水印

2.2.19.1 结构说明

水印（watermark）是文档中的背景文字或者图像，用于说明文档版权、防止文档侵权的特殊标记。水印仅可作为子要素存在页面（page）要素中，水印节点字段说明如下：

watermark			
字段名称	字段类型	是否必须	说明
id	String	是	编号
type	String	是	类型，填写watermark，表示水印节点
angle	Float	是	v1.1版本新增： 角度，取值范围[0, 360)，按照顺时针方向
coord	Point[]	是	水印的位置，是水印对应区域的最大外接矩形坐标
contour	Point[]	是	水印的轮廓，是水印对应区域的轮廓坐标
category	String	是	水印的类别： text: 文字水印 graph: 插图水印 compound: 混合水印 other: 其他
content	Object[][]	是	子要素集合，子要素仅可包含段落（paragraph）、插图（graph）、公式（formula）要素。其中，数组的第一个维度表示多候选，第二个维度表示子节点集合；

2.2.19.2 水印示例

下图给出了一个含有三个水印的图片示例，水印 1 类别为混合水印，水印 2 和水印 3 类别为文字水印。

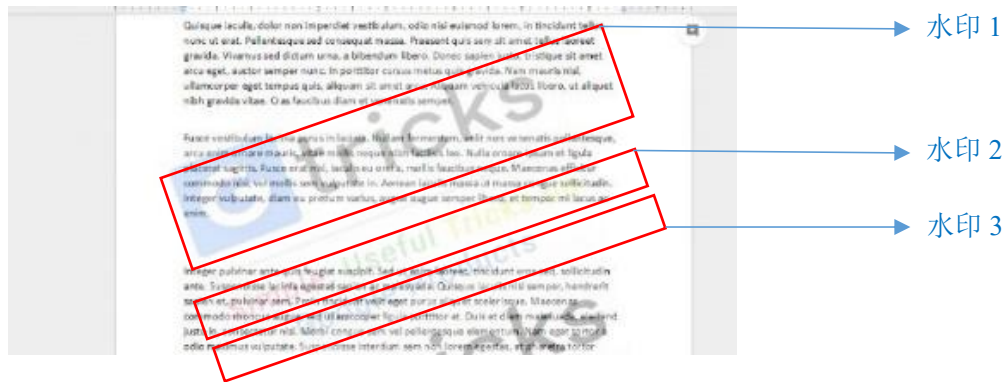


图 2-2-19-2-1：水印示例

上图对应 JSON 结果如下：

```
{
  "id": "xxx",
  "type": "page",
  "content":
  [
    // 水印 1
    {
      "id": "1",
      "type": "watermark",
      "coord": [...],
      "contour": [...],
      "category": "compound",
      "content":
      [
        // top1
        [
          {
            "id": "xxx",
            "type": "graph",
            "coord": [...],
            "contour": [...],
            "content": [[...]],
            "note": [...]
          },
          {
            "id": "xxx",
            "type": "paragraph",
            "attribute": [...],
            "coord": [...],
            "contour": [...],
            "content": [[...]]
          }
        ],
        [...],
        [...]
      ]
    },
    // 水印 2
    {
      "id": "2",
      "type": "watermark",
      "coord": [...],
```

```
"contour": [...],
"category": "text",
"content":
[
  // top1
  [
    {
      "id": "xxx",
      "type": "paragraph",
      "attribute": [...],
      "coord": [...],
      "contour": [...],
      "content": [[...].]
    }
  ],
  [...],
  [...]
],
// 水印 3
{
  ...
}
]
```

2.2.20 页脚

2.2.20.1 结构说明

页脚（page_footer）一般位于文档图像的底部区域，通常用来显示页码、版权信息、书名或其他出版信息。页脚示例如下图所示：

Q1 (SC: 0.23-0.72 mg/dL) (n = 923)				
OXD-EPI	20.57 (19.30, 21.78)*	24.68 (22.96, 26.52)	59.9 (55.0, 65.1)*	27.61 (26.58, 28.67)*
FAS	27.23 (25.14, 28.58)*	33.51 (30.43, 36.60)*	49.7 (45.3, 54.5)*	40.63 (38.51, 42.82)*
BRFC	14.01 (12.75, 15.27)	24.29 (22.43, 26.23)	72.6 (67.2, 78.3)	21.98 (21.07, 22.91)
Q2 (SC: 0.72-0.91 mg/dL) (n = 923)				
OXD-EPI	13.77 (12.39, 14.85)*	21.75 (20.14, 23.55)	69.6 (64.3, 75.2)	22.14 (21.20, 23.07)*
FAS	10.18 (8.97, 11.74)	24.45 (22.50, 26.37)*	74.1 (68.7, 79.9)	21.64 (20.64, 22.63)*
BRFC	8.74 (7.29, 10.04)	21.56 (19.96, 23.35)	77.1 (71.6, 83.0)	19.24 (18.38, 20.18)
Q3 (SC: 0.91-1.19 mg/dL) (n = 923)				
OXD-EPI	5.77 (4.81, 7.23)*	20.16 (18.73, 21.58)	78.7 (73.0, 84.6)	17.16 (16.27, 18.12)*
FAS	2.33 (1.36, 3.60)	20.22 (18.61, 21.78)	81.4 (75.7, 87.4)	16.18 (15.29, 17.16)
BRFC	2.98 (1.91, 4.28)	19.94 (18.69, 21.40)	80.9 (75.2, 87.0)	16.29 (15.40, 17.18)
Q4 (SC: 1.19-18.99 mg/dL) (n = 923)				
OXD-EPI	-7.46 (-8.38, -6.38)	20.19 (18.62, 21.85)*	61.9 (56.9, 67.2)	17.91 (16.97, 18.93)*
FAS	-6.18 (-7.61, -5.28)	18.24 (16.89, 19.91)*	64.8 (59.7, 70.2)	16.89 (15.97, 17.88)*
BRFC	-7.69 (-9.16, -6.58)	19.23 (17.77, 20.90)	62.6 (57.6, 67.9)	17.80 (16.89, 18.76)
*p < 0.05, compared with BRFC				

68	Nephren 2024/14835-73 DOI: 10.1191000031000	Ma Wei Tong Yu Chen Zhai Zhu
----	--	------------------------------

→ 页脚

图 2-2-20-1-1：页脚示例

页脚仅可作为子要素存在于页面 page 中，页脚具体字段说明如下：

page_footer			
字段名称	字段类型	是否必须	说明
id	String	是	编号
type	String	是	类型，填写page_footer，表示页脚要素

angle	Float	是	v1.1版本新增: 角度，取值范围[0, 360)，按照顺时针方向
coord	Point[]	是	页脚的位置，是页脚对应区域的最大外接矩形坐标
contour	Point[]	是	页脚的轮廓，是页脚对应区域的轮廓多边形坐标
content	Object[][]	是	页脚子要素集合，子要素仅可包括段落（paragraph）、表格（table）、插图（graph）、页码（page_number）、公式（formula）、列表（list）、二维码（qrcode）、条形码（barcode）、键值对（key和value）要素。其中，数组的第一个维度表示多候选，第二个维度表示子节点集合

2.2.20.2 页脚示例

下图给出了一个页脚示例：

参考文献

- [1] PERES M A, MACPHERSON L M D, WEYANT R J, et al. Oral diseases: a global public health challenge[J]. Lancet, 2019, 394(10194): 249-260.
- [2] PETERSEN P E, OGAWA H. The global burden of periodontal disease: towards integration with chronic disease prevention and control [J]. Periodontol 2000, 2012, 60(1): 15-39.
- [3] SUN H Y, JIANG H, DU M Q, et al. The prevalence and associated factors of periodontal disease among 35 to 44-year-old Chinese adults in the 4th national oral health survey [J]. Chin J Dent Res, 2018; 21(4): 241-247.
- [4] JIAO J, JING W D, SI Y, et al. The prevalence and severity of periodontal disease in Mainland China: data from the Fourth National Oral Health Survey (2015-2016)[J]. J Clin Periodontol, 2021, 48(2): 168-179.
- [5] LERTPIMONCHAI A, RATTANASIRI S, ARJONG VALLIBHAKARA S, et al. The association between oral hygiene and periodontitis: a systematic review and meta-analysis[J]. Int Dent J, 2017, 67

<http://www.murxiv.org.cn>

→ 页脚

图 2-2-20-2-1：页脚示例

上图中还包含段落等要素，为了便于说明，图中仅标注了页脚要素，上图对应的 JSON 结果如下：

```
{
  "type": "page",
  ...
  "content":
  [
    [
      // 页脚
    ]
  ]
}
```

```
"type": "page_footer",
...
"content":
[
  // 多候选 top1
  [
    {
      "type": "paragraph",
      ...
      "content":
      [
        [
          {
            "type": "textline",
            ...
            "content": [[]],
            "text": ["http://www.nursxiv.com"]
          }
        ]
      ],
      "text": ["http://www.nursxiv.com"]
    }
  ],
  // 多候选 top2
  [...]
]
}
```

2.2.21 页码

2.2.21.1 结构说明

页码（page_number）是指文档中用于标识文档页面编号的区域。页码仅可作为子要素存在于页眉（page_header）、页脚（page_footer）和页面（page）要素中，如果页面上没有页眉、页脚，则页码直接作为页面的子要素，存放于页面的内容（content）节点下，页码节点字段说明如下：

page_pumber			
字段名称	字段类型	是否必须	说明
id	String	是	编号
type	String	是	类型，填写page_numder，表示页码要素
angle	Float	是	v1.1版本新增： 角度，取值范围[0, 360)，按照顺时针方向
coord	Point[]	是	页码的位置，是页码对应区域的最大外接矩形坐标

contour	Point[]	是	页码的轮廓，是页码对应区域的轮廓多边形坐标
content	Paragraph[][]	是	子要素集合，子要素仅可包含段落（paragraph）、插图（graph）、公式（formula）要素。其中，数组的第一个维度表示多候选，第二个维度表示子节点集合；

下面给出了一个包含页码要素的图片示例：

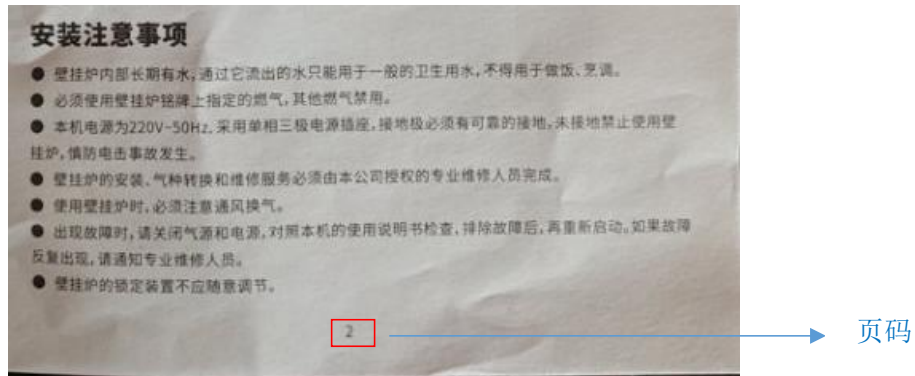


图 2-2-21-1-1：页码示例

2.2.21.2 页码示例

下面给出了一个包含页码要素的图片示例：

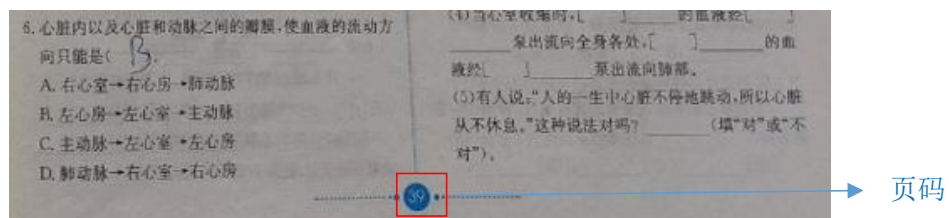


图 2-2-21-2-1：页码示例

上图中除包含页码以外，还包含分栏、段落等要素，为了便于说明，图中仅标注出了页码要素，上图对应 JSON 结果如下：

```
{
  "type": "page",
  ...,
  "content":
  [
    [
      // 页码
      {
        "type": "page_number",
        ...,
        "content":
        [
          // 多候选 top1
          [
```

```
{
  "type": "paragraph",
  "...",
  "content": [[...]],
  "text": ["39"]
},
// 多候选 top2
[...],
// 多候选 top3
[...],
]
}
]
}
]
```

2.2.22 批注

2.2.22.1 结构说明

批注（annotation）是对文档内容的修订或者说明，例如试卷上的分数，作业的对号和叉号，以及在书本上书写的笔记。批注要素字段说明如下：

annotation			
字段名称	字段类型	是否必须	说明
id	String	是	编号
type	String	是	类型，填写annotation，表示批注要素
angle	Float	是	v1.1版本新增： 角度，取值范围[0, 360)，按照顺时针方向
coord	Point[]	是	批注的位置，是批注对应区域的最大外接矩形坐标
contour	Point[]	是	批注的轮廓，是批注对应区域的轮廓多边形坐标
content	Paragraph[][]	是	批注子要素集合，子要素仅可包括段落（paragraph）、公式（formula）、表格（table）要素。其中，数组的第一个维度表示多候选，第二个维度表示子节点集合

2.2.22.2 批注示例

下图给出一个批注示例：

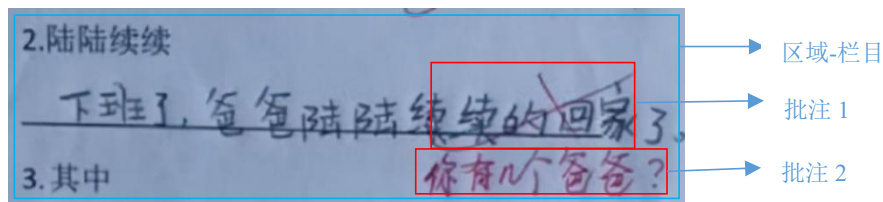


图 2-2-22-2-1：批注示例

上图中包含一个类别为栏目的区域，区域内包含两个批注子要素，表示批注内容归属于此区域。上图对应的 JSON 结果如下：

```
{
  // 区域-栏目
  "id": "xx",
  "type": "region",
  "coord": [],
  "contour": [],
  "category": "column",
  "attribute": [],
  "content":
  [
    [
      // 批注一
      {
        "id": "xx",
        "type": "annotation",
        "coord": [],
        "contour": [],
        "content":
        [
          [
            {
              "id": "",
              "type": "paragraph",
              "...",
              "text": ["\\checkmark"]
            }
          ]
        ]
      }
    ]
  ]
  // 批注二
  {
    "id": "xx",
    "type": "annotation",
    "coord": [],
    "contour": [],
    "content":
    [
      [
        {
          "id": "",
          "type": "paragraph",
          "...",
          "text": ["你有个爸爸? "]
        }
      ]
    ]
  }
]
}
```

2.2.23 脚注

2.2.23.1 结构说明

脚注（footnote）通常位于页面的底部，用于对文中的某一部分进行补充说明，与页脚不同的是，脚注左上角通常带有数字序号。脚注仅可作为子要素存在于区域（region）要素中。脚注字段的说明如下：

footnote			
字段名称	字段类型	是否必须	说明
id	String	是	编号
type	String	是	类型，填写footnote，表示脚注要素
angle	Float	是	v1.1版本新增： 角度，取值范围[0, 360)，按照顺时针方向
coord	Point[]	是	脚注的位置，是脚注对应区域的最大外接矩形坐标
contour	Point[]	是	脚注的轮廓，是脚注对应区域的轮廓多边形坐标
content	Object[][]	是	脚注子要素集合，子要素仅可包括段落（paragraph）、插图（graph）、公式（formula）、表格（table）、列表（list）、键值对（key和value）、二维码（qrcode）、条形码（barcode）要素。其中，数组的第一个维度表示多候选，第二个维度表示子节点集合

2.2.23.2 脚注示例

下面给出了一个包含脚注要素的图片示例：



图 2-2-23-2-1：脚注示例

上图中还包含段落等要素，为了便于说明，图中仅标出脚注要素。上图对应的 JSON 结果如下：

```
{
  "type": "region",
  "category": "column",
  "content": [
    [
      // 脚注
      {
        "type": "footnote",
        "content": [
          // 多候选 top1
          [
            {
              "type": "paragraph",
              "text": ["是一家主营在线教育和教育电子类..."]
            },
            {
              "type": "paragraph",
              "text": ["Word\\PPT..."]
            }
          ],
          // 多候选 top2
          [...]
        ]
      }
    ]
  ]
}
```

2.2.24 键值对

2.2.24.1 结构说明

键值对（key 和 value）是指文档中存在 key-value 结构化关系的文字，如身份证中的“姓名”与“张三”、“性别”与“男”。键值对仅可作为子要素存在于版面（layout）、区域（region）和单元格（cell）要素中，分别以 key 节点或 value 节点的形式存储，节点字段说明如下：

key			
字段名称	字段类型	是否必须	说明
id	String	是	编号
type	String	是	类型，取值key，表示键
angle	Float	是	角度，取值范围[0, 360)，按照顺时针方向

coord	Point[]	是	键（key）的位置，为键（key）区域的最大外接矩形坐标
contour	Point[]	是	键（key）的轮廓，是键（key）区域的轮廓坐标
content	Object[][]	是	key的具体内容，仅可包含段落（paragraph）、公式（formula）、插图（graph）、列表（list）要素。其中，数组的第一个维度表示多候选（但对于key来说，只存在top1候选），第二个维度表示子节点集合
key_group	String[]	是	属于同一个key的ID列表
value_group	String[]	是	key关联的value节点的ID列表

value			
字段名称	字段类型	是否必须	说明
id	String	是	编号
type	String	是	类型，取值value，表示键对应的值
angle	Float	是	角度，取值范围[0, 360)，按照顺时针方向
coord	Point[]	是	值（value）的位置，为值（value）区域的最大外接矩形坐标
contour	Point[]	是	值（value）的轮廓，是值（value）区域的轮廓坐标
content	Object[][]	是	value的具体内容，仅可包含段落（paragraph）、公式（formula）、插图（graph）、列表（list）要素。其中，数组的第一个维度表示多候选，第二个维度表示子节点集合
key_group	String[]	是	属于同一个key的ID列表
value_group	String[]	是	key关联的value节点的ID数组

2.2.24.2 键值对示例

下图给出了一个常见的键值对示例：

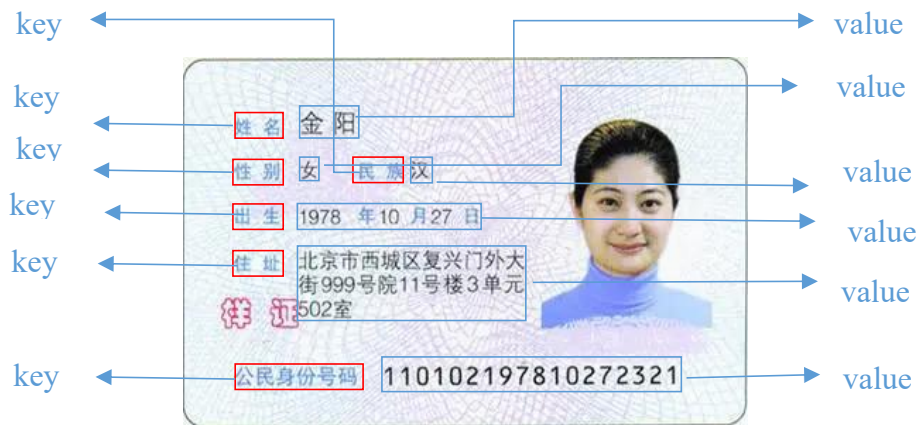


图 2-2-24-2-1：键值对示例

上图中除了包含键值对要素以外，还包含段落、插图等要素，为了便于说明，图中仅标注了键值对要素，上图对应的 JSON 结果如下：

```
{
  ...
  {
    "type": "page",
    ...
    "content":
    [
      // 版面
      {
        "type": "layout",
        ...
        "content":
        [
          // 多候选 top1
          [
            // 键 (key)
            {
              "id": "1",
              "type": "key",
              ...
              "content":
              [
                // 段落
                {
                  "type": "paragraph",
                  "content":
                  [
                    {
                      "type": "textline",
                      ...
                    }
                  ],
                  "text": ["姓 名"]
                }
              ]
            },
            "key_group": ["1"],
            "value_group": ["2"]
          ],
          // 值 (value)
          {
            "id": "2",
            "type": "value",
            ...
          }
        ]
      }
    ]
  }
}
```

```
    "content":
      [
        [
          {
            "type": "paragraph",
            "content":
              [
                {
                  "type": "textline",
                  ...
                }
              ],
            "text": ["金 阳"]
          }
        ]
      ],
      "key_group": ["1"],
      "value_group": ["2"]
    }
    // 其他键值对
    {...}
  ],
  // 多候选 top2
  [...]
]
}
}
```

对于键（key）为空的值（value），或者无法被进行结构化的文本，会当作正常文本直接放置于段落（paragraph）中，而不放在 key 和 value 节点中。如下图所示的“设备采购合同”与“（必须填写公司全称）”均为无法进行 key-value 结构化的文本。

设备采购合同

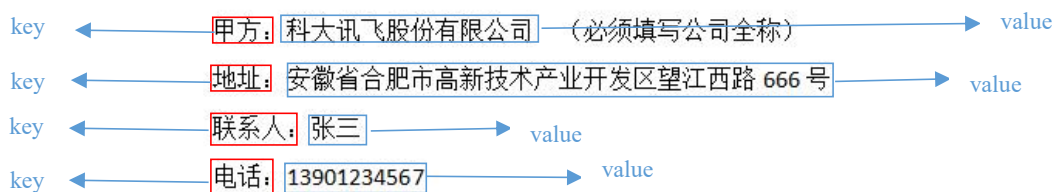


图 2-2-24-2-2：键值对示例

上图对应的 JSON 结果示例如下：

```
{
  ...
  {
    "type": "page",
    ...
    "content":
      [

```

```
// 版面
{
  "type": "layout",
  "...",
  "content":
  [
    // 多候选 top1
    [
      // 区域 (region)
      {
        "id": "1",
        "type": "region",
        "category": "column",
        "attribute": [],
        "content":
        [
          [
            {
              "id": "2",
              "type": "paragraph",
              "content":
              [
                {
                  "type": "textline",
                  ...
                }
              ],
              "text": ["设备采购合同"]
            }
          ]
        ]
      }
    ]
  ]
  // 区域 (region)
  {
    "id": "3",
    "type": "region",
    "category": "column",
    "attribute": [],
    "content":
    [
      [
        {
          "id": "4",
          "type": "paragraph",
          "content":
          [
            {
              "type": "textline",
              ...
            }
          ],
          "text": ["（必须填写公司全称）"]
        }
      ]
    ]
  }
  // 键 (key)
  {
    "id": "5",
    "type": "key",
    "...",
    "content":
    [
      [
        // 段落
        {
          "type": "paragraph",
          "content":
          [
            {
              "type": "textline",
              ...
            }
          ],
          ...
        }
      ]
    ]
  }
}
```

```
        "text":["甲方: "]
      }
    ],
    "key_group": ["5"],
    "value_group": ["6"]
  },
  // 值 (value)
  {
    "id": "6",
    "type": "value",
    ...,
    "content":
    [
      [
        {
          "type": "paragraph",
          "content":
          [
            {
              "type": "textline",
              ...
            }
          ],
          "text":["科大讯飞股份有限公司"]
        }
      ]
    ]
    "key_group": ["5"],
    "value_group": ["6"]
  }
  // 其他键值对
  {...}
],
// 多候选 top2
[...]
```

当前暂不支持对表格中的键值对进行结构化，但未来会持续更新并支持。如果 key 和 value 跨单元格，则会将 key 和 value 分别放置于对应的单元格(cell)内，如下图所示。

第一人民医院 体检登记表

体检号:	20121008	姓名:	张三	职业:	工程师
年龄:	23 岁	性别:	男	婚姻状况:	未婚

图 2-2-24-2-3: 键值对跨单元格的示例

上图对应的 JSON 结果示例如下，仅展示表格中跨单元格的 key 和 value 要素：

```
{
  // 表格
  "type": "table",
  "col": 6,
  "row": 2,
  ...,
  "category": "full_line",
  "cells":
  [
    // 单元格 1
    {
      "type": "cell",
      "category": "normal",
      ...,
      "content":
      [
        // 键 (key)
        {
          "id": "1",
          "type": "key",
          ...,
          "content":
          [
            [
              // 段落
              {
                "type": "paragraph",
                "content":
                [
                  {
                    "type": "textline",
                    ...
                  }
                ]
              },
              "text": ["体检号: "]
            ]
          ],
          "key_group": ["1"],
          "value_group": ["2"]
        },
      ],
    },
    // 单元格 2
    {
      "type": "cell",
      "category": "normal",
      ...,
      "content":
      [
        // 值 (value)
        {
          "id": "2",
          "type": "value",
          ...,
          "content":
          [
            [
              // 段落
              {
                "type": "paragraph",
                "content":
                [
                  {
                    "type": "textline",
                    ...
                  }
                ]
              },
              "text": ["20121008"]
            ]
          ]
        }
      ],
    }
  ]
}
```

```

    ],
    "key_group": ["1"],
    "value_group": ["2"]
  },
} ]
// 其他单元格
{...}
]
}

```

2.2.25 目录

2.2.25.1 结构说明

目录（contents）是指书籍正文前的章节导航，在书籍、报告或其他印刷品中，目录列出了书籍或报告中主要章节的标题，以及它们对应的页码。节点字段说明如下：

contents			
字段名称	字段类型	是否必须	说明
id	String	是	编号
type	String	是	类型，取值contents，表示目录要素
angle	Float	是	角度，取值范围[0, 360)，按照顺时针方向
attribute	Object[]	否	区域的属性集合
coord	Point[]	是	目录的位置，是目录对应区域的最大外接矩形坐标
contour	Point[]	是	目录的轮廓，是目录行对应区域的轮廓坐标
content	Object[][]	是	目录的具体内容，仅可包含段落(paragraph)、公式(formula)、插图(graph)、标题(title)要素。其中，数组

			的第一个维度表示多候选，第二个维度表示子节点集合。
--	--	--	---------------------------

2.2.25.2 目录属性

通用场景中，目录属性（attribute）字段取值说明如下：

属性	字段名	字段类型	字段说明
关联要素	name	String	relation
	value	String[]	存放属于同一目录的要素的编号（id）

当目录存在跨栏跨页情况时，每个区域都会赋予关联要素（relation）属性，属性的 value 字段取值为属于同一目录的 id 集合。

2.2.25.3 目录示例

下图给出了一个目录示例：

目 录	
绪论.....	(1)
第一章 流体力学基础.....	(6)
1.1 概述	(6)
1.2 流体静力学及其应用	(8)
1.2.1 静止流体所受的力	(8)
1.2.2 流体静力学基本方程	(9)
1.2.3 静力学原理在压力和压力差测量上的应用.....	(11)
1.3 流体流动的基本方程.....	(14)
1.3.1 基本概念.....	(14)
1.3.2 质量衡算方程——连续性方程.....	(20)
1.3.3 运动方程.....	(24)
1.3.4 总能量衡算和机械能衡算方程.....	(34)
1.4 管路计算.....	(53)
1.4.1 简单管路.....	(53)
1.4.2 复杂管路.....	(58)
1.4.3 管网简介.....	(64)
1.4.4 可压缩流体的管路计算.....	(64)
1.5 边界层及边界层方程.....	(68)
1.5.1 普兰特边界层理论.....	(68)
1.5.2 普兰特边界层方程.....	(70)
1.5.3 边界层积分动量方程及其应用.....	(72)
1.5.4 边界层分离.....	(77)
1.6 湍流.....	(78)
1.6.1 湍流特点及其研究方法.....	(78)
1.6.2 湍流应力.....	(80)
1.7 流速、流量测量	(82)
1.7.1 变压头流量计.....	(82)
1.7.2 变截面流量计.....	(86)
本章主要符号说明	(89)

→ 目录

图 2-2-25-3-1：目录

上图中，包含一个目录要素，其中目录要素内部包含文字，图像内的文字均纳入目录子要素中，以段落的形式组织输出，上图对应 JSON 结果如下：

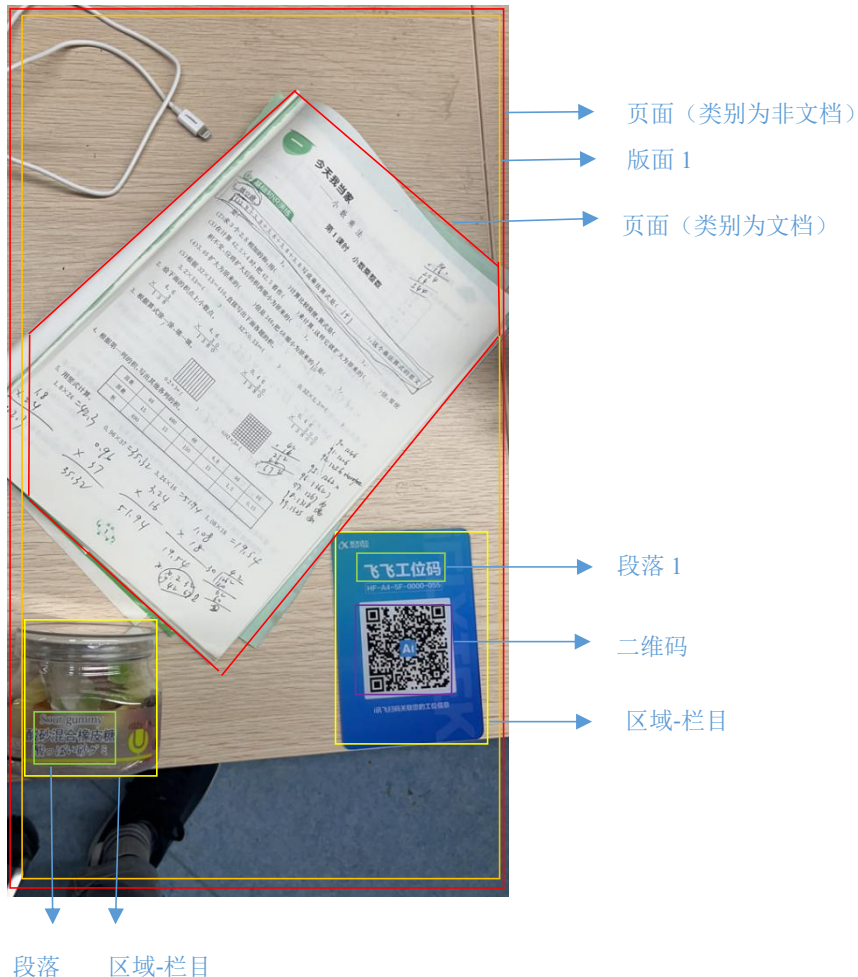
```
{
  "type": "region",
  "category": "column",
  "content":
  [
    [
      // 目录
      {
        "type": "contents",
        "id": "12",
        "attribute": [],
        "content":
        [
          [
            // 图像内文字
            {
              "type": "paragraph",
              "content": [[...]],
              "text": ["目录"]
            },
            {
              "type": "paragraph",
              "content": [[...]],
              "text": ["绪论"]
            },
            {
              "type": "paragraph",
              "content": [[...]],
              "text": ["第一章 流体力学基础"]
            }
          ],
          [...],
          [...]
        ],
        "note": [...]
      },
    ]
  ]
}
```

2.3 特殊场景示例

通用 OCR 大模型输出协议涉及到多种要素之间互相嵌套和组合，为更好的描述和表示要素之间的关系，本章节会针对性给出部分特殊场景的相关图例和 JSON 示例。

2.3.1 文档与非文档混合场景

下图给出了一个文档页面和非文档页面混合的图片样例：



上图是一个文档图像，图像包含包含一个文档页面和一个非文档页面，图像中出文档区域以外的区域作为一个非文档页面，文档区域作为一个文档页面。页面下包含版面，二维码因为是独立要素所以存放在版面外，为了便于说明，上图中段落要素下文本行、文本单元等要素未标注。上图对应 JSON 结果如下：

```
{
  "engine_version": "",
  "version": "",
  "document": [],
  "image": [
    [
      // 非文档页面
      {
        "id": "1",
        "width": 80,
        "attribute": [],
        "height": 100,
        "content": [
          [
```

```
// 非文档页面
{
  "id": "2",
  "category": "other",
  "type": "page",
  ...
  "content":
  [
    [
      // 非文档页面下浮动要素二维码
      {
        "id": "3",
        "type": "qrcode",
        ...
        "content": [[]]
      }
    ]
    // 非文档页面下版面 1
    {
      "id": "4",
      "type": "layout",
      ...
      "content":
      [
        [
          // 区域-栏目
          {
            "id": "5",
            "type": "region",
            ...
            "category": "column",
            "content":
            [
              [
                // 段落 1
                {
                  "id": "6",
                  "type": "paragraph",
                  ...
                  "content":
                  [
                    [
                      // 文本行
                      {
                        "id": "7",
                        "type": "textline",
                        ...
                        "direction": "horizontal",
                        "attribute":
                        [
                          {
                            "name": "indent",
                            "value": 0
                          },
                          {
                            "name": "alignment",
                            "value": "center"
                          }
                        ]
                      },
                      "content":
                      [
                        [
                          // 文本块
                          {
                            "type": "text_block",
                            "category": "text",
                            ...
                            "content":
                            [
                              [
                                // 文本单元
                                {
                                  ...
                                  "type": "text_unit",
                                  "category": "print",
                                  "attribute": [...],
                                  "text": "飞飞工位码",
                                }
                              ]
                            ]
                          }
                        ]
                      ]
                    ]
                  ]
                }
              ]
            ]
          }
        ]
      ]
    }
  ]
}
```

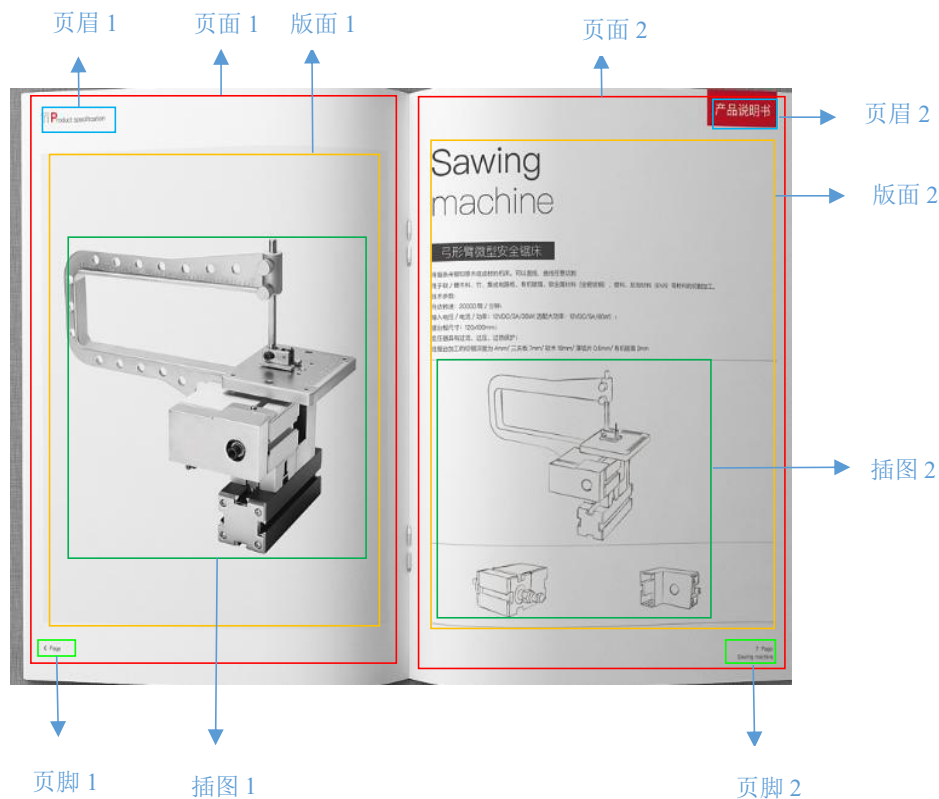
```

        "word":
        [
            // 多候选 top3
            [...],
            [...],
            [...]
        ],
        "char":
        [
            // 多候选 top3
            [...],
            [...],
            [...]
        ]
    }
},
"text":["飞飞工位码","...","..."]
}],
"text":["飞飞工位码","...","..."]
},
{...}
}
]
},
{...}
}
}
}
}
// 文档页面
{
    "type": "page",
    "category": "document",
    "...",
    "content":
    [
        [
            // 文档页面版面
            {
                "type": "layout",
                "...",
                "content": [[...]]
            }
        ]
    ]
}
}
}
}
}

```

2.3.2 产品文档场景

下图给出了一个包含产品文档的图片示例：



上图分为两个页面，左边页面由页眉、页脚和单个区域组成；右边页面由标题、段落、插图、页脚组成，其中页脚要素包含页码、段落。图中还包含段落、文本行等要素，为了便于说明，图中未标注出，上图对应的 JSON 结果如下：

```
{
  "engine_version": "",
  "version": "",
  "document": [],
  "image": [
    {
      "id": "",
      "width": 2400,
      "attribute": [],
      "height": 2200,
      "content": [
        [
          // 页面 1
          {
            "id": "",
            "type": "page",
            "category": "document",
            "content": [
              [
                // 版面
                {
                  "id": "",
                  "type": "layout",
                  "content": [
                    [
                      // 插图 1
```

```

        {
          "id": "",
          "type": "graph",
          "coord": [],
          "contour": [],
          "notes": [],
          "content": [[...]]
        }
      ]
    },
    // 页码 1
    {
      "id": "xxx",
      "type": "page_numder",
      "...",
      "content":
      [
        [
          {
            "id": "",
            "type": "paragraph",
            "...",
            "content":
            [
              [
                {
                  "id": "",
                  "type": "textline",
                  "direction": "horizontal",
                  "...",
                  "content": [[...]],
                  "text": ["6 Page"]
                }
              ]
            ],
            "text": ["6 Page"]
          }
        ]
      ]
    },
    // 页眉
    {
      "id": "",
      "type": "page_header",
      "...",
      "content": [[...]]
    }
  ]
},
// 页面 2
{
  "id": "",
  "type": "page",
  "category": "document",
  "...",
  "content":
  [
    [
      // 版面 2
      {
        "id": "",
        "type": "layout",
        "...",
        "content":
        [
          [
            // 页眉 2
            {
              "id": "",
              "type": "page_header",
              "...",
              "content":
              [

```

```

[
  {
    "type": "paragraph",
    "content":
    [
      [
        {
          "type": "textline",
          "content": [...],
          "text": ["产品说明书"]
        }
      ]
    ],
    "text": ["产品说明书"]
  }
]
// 插图 2
{
  "id": "",
  "type": "graph",
  "coord": [],
  "contour": [],
  "notes": [],
  "content": [...]
}
]
// 页脚
{
  "type": "page_footer",
  "content":
  [
    [
      // 页码
      {
        "type": "page_number",
        "content": [...]
      },
      // 段落
      {
        ...
      }
    ]
  ]
}

```

2.4 常见问题说明

2.4.1 表格嵌套表格识别

引擎暂不支持表格嵌套表格识别，即单元格内的子表格不识别，仅识别子表格内的文字，丢弃表格结构。

2.4.2 要素输出排序

引擎输出的各类要素按照阅读顺序输出的，其中要素的 `content` 节点的第一维是表示多候选，第一候选存放在序号为 0 的数组中，第二候选存放在序号为 1 的数组中，以此类推。

3、Markdown 输出协议

3.1 协议概述

Markdown 输出协议描述了图像识别结果的 Markdown 格式输出规范，此规范详细说明了标题、段落、表格、列表、代码、公式等要素的表示、输出和展示效果。本文档使用 Visual Studio Code 的 Markdown All in One 插件展示 Markdown 渲染效果。Markdown 结果以 Markdown 格式进行封装，保存在输出结果 `document` 节点下，具体参见第 2.2 节协议说明。

3.2 协议说明

在介绍 Markdown 输出协议之前，首先给出一个简单的示例说明 Markdown 输出规则和展示效果。



图 3-2-1：图片识别结果示意图

上图中，① - ⑧表示阅读顺序，此顺序由引擎输出。引擎不输出要素中的图片，但是如果要素中包含文字，则输出该文字信息，上图的插图④仅包含图片，引擎不输出该要素。如果一个段落跨两个及以上的区域，引擎会将各子段落合并成一个完整的段落进行输出，上图的 ⑥、⑦子段落属于一个段落，引擎会将其合并输出。上图对应的 Markdown 原文本及渲染效果如下所示：

表 3-2-1：示例图片的 Markdown 输出及渲染效果

Markdow原文本	Markdown渲染效果
------------	--------------

2017 年如何紧跟大势，找准投资方向
——我会联合主办“2017 年货币走向与财富管理”主题论坛

过去的 2016 年，从英国脱欧到美国大选，从美联储 12 月加息到人民币贬值，国际局势“黑天鹅”频繁起飞。2016 年全球经济跌宕起伏，各大市场频繁上演“过山车”行情，让无数人跌破眼镜，展望 2017 年，投资者们将更加如履薄冰。面对巨大的不确定性，如何在 2017 年找准投资方向，紧跟财富管理趋势？

对于全球金融市场来说，2016 年注定不平凡，意外连连。从英国脱欧、美国大选到意大利公投，从美联储 12 月加息到人民币贬值，金融市场在 2016 年经历了“黑天鹅”的频繁起飞，更遭受着其引发的巨大冲击。

“黑天鹅”代表着不可预测的重大稀有事件，它在意料之外。却又瞬间改变一切。在投资市场上，“黑天鹅”的出现，往往意味着难以预测而又重大的损失。2016 年的会融形势让无数人跌破眼睛，展望 2017 年。投资者们将更加如履薄冰。

面对巨大的不确定性，如何在 2017 年找准投资方向，紧跟财富管理趋势，为了寻找答案，深圳市前海金融创新促进会携手香港大学 SPACE 中国商学院、前海产业智库共同邀请摩根大通、花旗银行、罗斯柴尔德家族银行亚洲区前董事邹宇先生、金融法律专家曾文生博士这两位资深业界专家为大家奉上了 2017 年开年第一场全融思维盛宴。

2017年如何紧跟大势，找准投资方向

——我会联合主办“2017年货币走向与财富管理”主题论坛

过去的2016年，从英国脱欧到美国大选，从美联储12月加息到人民币贬值，国际局势“黑天鹅”频繁起飞。2016年全球经济跌宕起伏，各大市场频繁上演“过山车”行情，让无数人跌破眼镜，展望2017年，投资者们将更加如履薄冰。面对巨大的不确定性，如何在2017年找准投资方向，紧跟财富管理趋势？

对于全球金融市场来说，2016年注定不平凡，意外连连。从英国脱欧、美国大选到意大利公投，从美联储12月加息到人民币贬值，金融市场在2016年经历了“黑天鹅”的频繁起飞，更遭受着其引发的巨大冲击。

“黑天鹅”代表着不可预测的重大稀有事件，它在意料之外。却又瞬间改变一切。在投资市场上，“黑天鹅”的出现，往往意味着难以预测而又重大的损失。2016年的会融形势让无数人跌破眼睛，展望2017年。投资者们将更加如履薄冰。

面对巨大的不确定性，如何在2017年找准投资方向，紧跟财富管理趋势，为了寻找答案，深圳市前海金融创新促进会携手香港大学SPACE中国商学院、前海产业智库共同邀请摩根大通、花旗银行、罗斯柴尔德家族银行亚洲区前董事邹宇先生、金融法律专家曾文生博士这两位资深业界专家为大家奉上了2017年开年第一场全融思维盛宴。

引擎支持标题、段落、表格、列表、代码、公式等 23 种要素识别，JSON 协议支持输出上述全部要素，但在 Markdown 协议中，为了方便用户使用，仅按照阅读顺序输出识别文字结果，文字结果以段落形式进行组织，暂不支持图像、文本行等要素输出。为了方便用户理解，我们将要素分为三类：可直接输出的要素，不可直接输出的要素、不输出的要素，见下表：

表 3-2-2：引擎输出要素分类表

类别	要素	说明
可直接输出的要素	标题、段落、表格、列表、代码段、公式	该要素节点包含文本信息，但是文本行要素例外，引擎不会直接输出文本行，会将文本行汇总成段落，以段落的形式进行输出
不可直接输出的要素	印章、信息栏、二维码、水印、条形码、页眉、页脚、页码、页面、版面、区域、插图、伪代码段、批注、脚注	该要素节点不包含文本信息，但该要素的子要素包含文本信息，文本信息以段落的形式进行输出
不输出的要素	指印、文本行	不可输出要素可细分为 2 种： (1) 该要素及子要素均不包含文本信息，例如指印 (2) 该要素及子要素的文本信息均不输出，例如文本行，文本行不单独输出，以段落的形式汇总输出

3.2.1 标题

引擎支持输出一至五级标题（标题划分标准参见 2.2.5 节），标题以“井号（#）+空格+标题名”格式表示，其中井号（#）是标题前导符，前导符与标题文字之间保留一个空格。以下是标题示例：

表 3-2-1-1：引擎输出的标题及渲染效果示例

Markdown原文本	Markdown渲染效果
# 一级标题	一级标题
## 二级标题	二级标题
### 三级标题	三级标题
#### 四级标题	四级标题
##### 五级标题	五级标题

标题的字体及字号大小由 Markdown 编辑器决定，一般情况下，字体为微软雅黑（部分编辑器会将标题加粗），四级标题和正文的字号大小相同，正文字号为 14px 或 16px。

本协议使用的编辑器（Markdown All in One）渲染标题时，字体为微软雅黑，字号大小与正文呈现倍数关系，正文大小为 14px，一级标题为正文的 2 倍（28px），二级标题为正文的 1.5 倍（21px），三级标题为正文的 1.17 倍（16.38px），四级标题为正文的 1 倍（14px），五级标题为正文的 0.83 倍（11.62px）。

3.2.2 段落

引擎输出的段落不需要任何特殊的标记，仅需在两个段落之间留出一个空行即可，以下为段落示例：

表 3-2-2-1：引擎输出的段落及渲染效果示例

Markdown原文本	Markdown渲染效果
人工智能是新一轮科技革命和产业变革的重要驱动力量，是研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门新的技术科学。 人工智能是智能学科重要的组成部分，它企图了解智能的实质，并生产出一种新的能以人类智能相似的方式做出反应的智能机器。	人工智能是新一轮科技革命和产业变革的重要驱动力量，是研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门新的技术科学。 人工智能是智能学科重要的组成部分，它企图了解智能的实质，并生产出一种新的能以人类智能相似的方式做出反应的智能机器。

引擎支持输出加粗文本，加粗文本以“*****+文本+*****”格式表示，以下给出一个普通文本和加粗文本同时输出的示例：

表 3-2-2-2：引擎输出的普通文本和加粗文本及渲染效果示例

Markdown原文本	Markdown渲染效果
** 人工智能是新一轮科技革命和产业变革的重要驱动力量 ** ，是研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门新的技术科学。 ** 人工智能是智能学科重要的组成部分，它企图了解智能的实质，并生产出一种新的能以人类智能相似的方式做出反应的智能机器。 **	人工智能是新一轮科技革命和产业变革的重要驱动力量，是研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门新的技术科学。 人工智能是智能学科重要的组成部分，它企图了解智能的实质，并生产出一种新的能以人类智能相似的方式做出反应的智能机器。

引擎支持输出斜体文本，斜体文本以“**+文本+**”格式表示，以下给出一个普通文本和斜体文本同时输出的示例：

表 3-2-2-3：引擎输出的普通文本和斜体文本及渲染效果示例

Markdown原文本	Markdown渲染效果
<i>*</i> 人工智能是新一轮科技革命和产业变革的重要驱动力量 <i>*</i> ，是研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门新的技术科学。 <i>*</i> 人工智能是智能学科重要的组成部分，它企图了解智能的实质，并生产出一种新的能以人类智能相似的方式做出反应的智能机器。 <i>*</i>	<i>人工智能是新一轮科技革命和产业变革的重要驱动力量，是研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门新的技术科学。</i> <i>人工智能是智能学科重要的组成部分，它企图了解智能的实质，并生产出一种新的能以人类智能相似的方式做出反应的智能机器。</i>

引擎支持输出斜体、加粗文本，斜体、加粗文本以“******+文本+******”格式表示，以下给出一个普通文本和斜体、加粗文本同时输出的示例：

表 3-2-2-4：引擎输出的普通文本和斜体、加粗文本及渲染效果示例

Markdown原文本	Markdown渲染效果
-------------	--------------

人工智能是新一轮科技革命和产业变革的重要驱动力量，是研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门新的技术科学。

人工智能是智能学科重要的组成部分，它企图了解智能的实质，并生产一种新的能以人类智能相似的方式做出反应的智能机器。

人工智能是新一轮科技革命和产业变革的重要驱动力量，是研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门新的技术科学。

人工智能是智能学科重要的组成部分，它企图了解智能的实质，并生产一种新的能以人类智能相似的方式做出反应的智能机器。

3.2.3 表格

引擎输出的表格基本结构由管道符（|）和连字符（-）组成。Markdown 原文本的每一行代表表格中的一行，而列则通过管道符分隔（两个管道符间无字符表示空单元格），原文本的第一行为表头，第二行为表格固定格式（渲染时不显示），从第三行开始为表体。以下为表格示例：

表 3-2-3-1：引擎输出的表格及渲染效果示例

Markdown原文本	Markdown渲染效果
星期一 星期二 星期三 星期四 星期五	
-- -- -- -- --	
1 数学 英语 数学 语文 品德	1 数学 英语 数学 语文 品德
2 美术 数学 品德 数学 语文	2 美术 数学 品德 数学 语文
3 品德 品德 语文 体育 数学	3 品德 品德 语文 体育 数学
4 语文 音乐 体育 品德 美术	4 语文 音乐 体育 品德 美术
5 数学 英语 语文 数学 语文	5 数学 英语 语文 数学 语文
6 品德 数学 美术 劳动 活动	6 品德 数学 美术 劳动 活动

引擎输出的表格支持三种列对齐方式，分别为左对齐、右对齐和居中对齐，其中左对齐使用“:--”表示（表格默认为左对齐），右对齐使用“--:”表示，居中对齐使用“:--:”表示。以下为表格居中对齐的示例：

表 3-2-3-2：引擎输出的列居中对齐表格及渲染效果示例

Markdown原文本	Markdown渲染效果
-------------	--------------

星期一 星期二 星期三 星期四 星期五	
:--: :--: :--: :--: :--:	
1 数学 英语 数学 语文 品德	
2 美术 数学 品德 数学 语文	
3 品德 品德 语文 体育 数学	
4 语文 音乐 体育 品德 美术	
5 数学 英语 语文 数学 语文	
6 品德 数学 美术 劳动 活动	

	星期一	星期二	星期三	星期四	星期五
1	数学	英语	数学	语文	品德
2	美术	数学	品德	数学	语文
3	品德	品德	语文	体育	数学
4	语文	音乐	体育	品德	美术
5	数学	英语	语文	数学	语文
6	品德	数学	美术	劳动	活动

引擎输出的表格可能会包含合并单元格，但是 Markdown 的表格不支持渲染合并单元格，针对上述情况，引擎在输出时，会将合并单元格拆分，并将合并单元格的内容复制到每个拆分后的子单元格中。下边给出一个简单的合并单元格拆分示例，其中图 3-2-3-1 为包含合并单元格的表格，表 3-2-3-3 为该表格对应的引擎输出和渲染效果。

合并单元格 1	←	项目		最小	典型	最大
合并单元格 2	←	电源参数	电压	--	12V	--
合并单元格 3	←		纹波	--	--	50mV
		电源电流	电流	3A		
			工作电流		250mA	380mA
			待机电流		22mA	30mA

图 3-2-3-1：包含合并单元格的表格示例

表 3-2-3-3：示例表格的 Markdown 输出及渲染效果

Markdown原文本	Markdown渲染效果
项目 项目 最小 典型 最大	
:--: :--: :--: :--: :--:	
电源参数 电压 -- 12V --	
电源参数 纹波 -- -- 50mV	
电源参数 电流 3A	
电源电流 工作电流 250mA 380mA	
电源电流 待机电流 22mA 30mA	

项目	项目	最小	典型	最大
电源参数	电压	--	12V	--
电源参数	纹波	--	--	50mV
电源参数	电流	3A		
电源电流	工作电流		250mA	380mA
电源电流	待机电流		22mA	30mA

3.2.4 列表

引擎支持输出有序列表和无序列表，其中有序列表是列表前带有数字编号的列表，无序列表是指列表前无数字，但带有诸如星号（*）、减号（-）等符号，转换后的列表项会丢弃前导符号，统一由编辑器处理（增加前导符）。有序列表、无序列表均支持多级列表嵌套。

3.2.4.1 有序列表

引擎输出的有序列表使用“数字+英文句号（.）+空格+列表内容”格式表示，多级嵌套列表需在列表项前添加四个空格，以下为有序列表的示例：

表 3.2.4.1.1：引擎输出的有序列表及渲染效果示例

Markdown 原文本	Markdown 渲染效果
学习目标 1. 了解人工智能的概念 2. 大致了解人工智能领域的发展历史 1. 人工智能的研究领域 1. 各领域的发展历史	学习目标 1. 了解人工智能的概念 2. 大致了解人工智能领域的发展历史 1. 人工智能的研究领域 1. 各领域的发展历史

3.2.4.2 无序列表

引擎输出的无序列表使用“星号（*）+空格+列表内容”格式表示，多级嵌套列表需在列表项前添加两个空格，以下为无序列表的示例：

表 3-2-4-2-1：引擎输出的无序列表及渲染效果示例

Markdown 原文本	Markdown 渲染效果
学习目标 * 了解人工智能的概念 * 大致了解人工智能领域的发展历史 * 人工智能的研究领域 * 各领域的发展历史	学习目标 • 了解人工智能的概念 • 大致了解人工智能领域的发展历史 ◦ 人工智能的研究领域 ▪ 各领域的发展历史

3.2.5 代码

引擎仅支持输出行间代码块，不支持输出行内代码，行间代码是指一行以上的代码段。代码的文字区域以三个反引号（`````）作为开始、结束的分隔符，并要求开始、结束的三个反引号单独成行，在开头的三个反引号后指定代码类型（引擎支持的代码类型见 2.2.12 节），例如 python 表示代码区域是 python 编写的代码。

表 3-2-5-1：引擎输出的行间代码及渲染效果示例

Markdown 原文本	Markdown 渲染效果
<pre>```python import os # 读取 demo.txt 文件 with open("demo.txt", "r") as f: lines = f.readlines() for line in lines: print(line) ... ```</pre>	

3.2.6 公式

引擎支持以 LATEX 格式输出公式识别结果，包括行内公式和行间公式，其中行间公式是单独成行的公式，行内公式是位于段落内的公式。LATEX 公式的表示及渲染效果见《LATEX 公式规范》。

3.2.6.1 行间公式

行间公式以两个美元符号（ $\\$$ ）作为前后分隔符，例如“ $\\$1+1=2\\$$ ”表示一个行间公式。当文本中出现“ $\\$$ ”符号时，使用“ $\\backslash\\$$ ”进行转义，即表示按照“ $\\$$ ”符的文字含义进行输出，例如“一瓶饮料 1 $\\backslash\\$$ ”。

表 3-2-6-1-1：引擎输出的行间公式及渲染效果示例

Markdown 原文本	Markdown 渲染效果
--------------	---------------

欧拉公式是一个著名的等式，可以表示为： \$\$ $e^{i\pi}+1=0$ \$\$	欧拉公式是一个著名的等式，可以表示为： $e^{i\pi} + 1 = 0$
---	---

3.2.6.2 行内公式

行内公式以一个美元符号（\$）作为前后分隔符，例如“\$1+1=2\$”表示一个行内公式。当文本中出现“\$”符号时，使用“\\$”进行转义，即表示按照“\$”符号的文字含义进行输出，例如“一台电脑 1000\\$”。

表 3-2-6-2-1：引擎输出的行内公式及渲染效果示例

Markdown原文本	Markdown渲染效果
欧拉公式可以表示为 $e^{i\pi}+1=0$ ，这是一个著名的等式。	欧拉公式可以表示为 $e^{i\pi} + 1 = 0$ ，这是一个著名的等式。

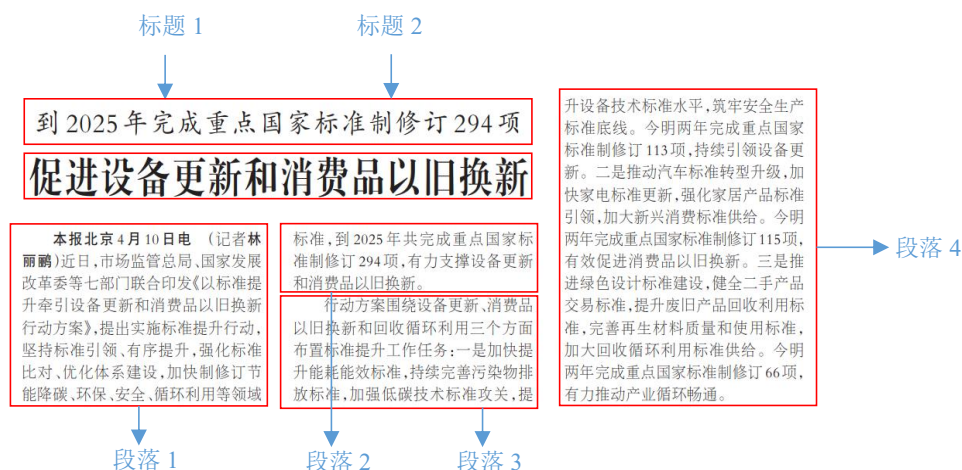
4、SED 输出协议

4.1 协议概述

SED（Simple Element Document）是简单要素格式的简称，采用简单要素格式输出时，识别的结果仅以段落和表格的形式进行组织，其他要素均转换为此两种格式表示。SED 结果以 JSON 格式进行封装，保存在输出结果 **document** 节点下，具体参见第 2.2 节协议说明。

4.2 协议说明

在介绍 SED 输出协议之前，首先给出一个简单的示例说明 SED 输出规则。



上图中，段落 1 和段落 2 是一个完整段落，段落 3 和段落 4 是一个完整段落，采用 SED 输出时，段落 1 和段落 2，段落 3 和段落 4 整合成 2 个完整段落输出，上图对应的 JSON 输出结果：

```
{
  "name": "sed",
  "value": [ {
    "type": "paragraph",
    "parent_type": "",
    "text": [ "到 2025 年完成重点国家标准制修订 294 项" ],
    "coord": [ [
      { "x": 100, "y": 100 },
      { "x": 100, "y": 200 },
      { "x": 200, "y": 200 },
      { "x": 100, "y": 200 }
    ]
  } ]
}
```

```
]
},
{
  "type": "paragraph",
  "parent_type": "",
  "text": [ "促进设备更新和消费品以旧换新" ],
  "coord": [ [ ... ] ]
},
{
  "type": "paragraph",
  "parent_type": "title",
  "text": [ "促进设备更新和消费品以旧换新" ],
  "coord": [ [ ... ] ]
},
{
  "type": "paragraph",
  "parent_type": "",
  "text": [
    "本报北京 4 月 10 日电(记者林丽鹂)近日,市场监管总局、国家发展改革委等七部门联合印
    发《以标准提升牵引设备更新和消费品以旧换新行动方案》提出实施标准提升行动坚持标准引领、有序
    提升,强化标准比对、优化体系建设,加快制修订节能降碳、环保、安全、循环利用等领域",
    "标准,到 2025 年共完成重点国家标准制修订 294 项,有力支撑设备更新和消费品以旧换
    新。"],
    "coord": [ [ ... ], [ ... ] ]
  },
  {
    "type": "paragraph",
    "parent_type": "",
    "text": [
      "行动方案围绕设备更新、消费品以旧换新和回收循环利用三个方面布置标准提升工作任
      务:一是加快提升能效能效标准,持续完善污染物排放标准,加强低碳技术标准攻关,提",
      "升设备技术标准水平,筑牢安全生产标准底线。今明两年完成重点国家标准制修订 113 项,
      持续引领设备更新。二是推动汽车标准转型升级,加快家电标准更新,强化家居产品标准引领,加大新兴
      消费标准供给。今明两年完成重点国家标准制修订 115 项有效促进消费品以旧换新。三是推进绿色设计
      标准建设,健全二手产品交易标准,提升废旧产品回收利用标准,完善再生材料质量和使用标准,加大回
      收循环利用标准供给。今明两年完成重点国家标准制修订 66 项,有力推动产业循环畅通。"],
      "coord": [ [ ... ], [ ... ] ]
    }
  ]
}
```

SED 协议具体说明:

字段名称	字段类型	是否必须	说明
type	String	是	表示要素类别,取值为: table: 表格 paragraph: 段落(行内公式用\$包围) code: 代码,类似"python XXX"格式 pseudocode: 伪代码 formula: 行间公式,用\$\$包围 上述要素去除样式标签
parent_type	String	否	表示该要素的父要素类型(当父节点是表格、段落时,父节点为空),可选值为: seal: 印章 information_bar: 信息栏 qrcode: 二维码 watermark: 水印 barcode: 条形码 page_header: 页眉 page_footer: 页脚 page_number: 页码 title: 标题

			table: 表格 graph: 插图 list: 列表 footnote: 脚注 formula: 行间公式 annotation: 批注 注意，以下要素不输出： 页面 (page)、版面 (layout)、区域 (region)、文本行 (textline)、指印 (fingerprint) 要素
text	string[]	是	text数组中的字符串表示段落的文字信息，通常情况下text数组仅包含一个字符串；当段落跨栏时，该数组会包含多个字符串，每个字符串表示一个子段落的文字信息)
coord	point[][]	是	text数组中的段落坐标集合。其中数组的第一个维度表示text数组的文本坐标数组，第二个维度表示text每个字符串对应区域的最大外接矩形坐标