Form-NLU：表单语言理解数据集

Yihao Ding 悉尼大学 悉尼，新南威尔士州，澳大利亚

Siqu Long 悉尼大学 悉尼，新南威尔士州，澳大利亚

Jiabin Huang 悉尼大学 悉尼，新南威尔士州，澳大利亚

Kaixuan Ren TheUniversityof 悉尼 悉尼，新南威尔士州，澳大利亚

Xingxiang Luo 悉尼大学 悉尼，新南威尔士州，澳大利亚

Hyunsuk Chung FortifyEdge 悉尼，新南威尔士州，澳大利亚

Soyeon Caren Han 悉尼大学 悉尼，新南威尔士州，澳大利亚

arXiv:2304.01577v3 [cs.IR] 3 Aug 2023

摘要

与通用文档分析任务相比，表单文档结构理解和检索颇具挑战。表单文档通常由两类人员制作：一类是表单设计者，负责设计表单结构和键；另一类是表单用户，根据提供的键填写表单值。因此，如果表单用户感到困惑，填写的表单值可能与表单设计者的意图（结构和键）不一致。本文介绍了 Form-NLU，这是首个用于表单结构理解及其键和值信息提取的新型数据集，旨在解读表单设计者的意图，以及用户填写的值与其意图之间的对应关系。该数据集包含 857 张表单图像、6 千个表单键和值，以及 4 千个表格键和值。数据集还涵盖三种表单类型：数字表单、印刷体表单和手写体表单，覆盖了多样的表单外观和布局。我们提出了一种基于位置和逻辑关系的稳健表单键值信息提取框架。利用 Form-NLU 数据集，我们首先考察了用于表单布局理解的强大目标检测模型，随后在该数据集上评估关键信息提取任务，并针对不同类型的表单和键提供细粒度的结果。此外，我们还使用现成的 PDF 布局提取工具对其进行考察，并证明了该工具在真实场景中的可行性。

CCS 概念

• **信息系统** →**信息检索**。

关键词

数据集、表单理解、自然语言理解

ACM 参考格式：

Yihao Ding、Siqu Long、Jiabin Huang、Kaixuan Ren、XingxiangLuo、Hyunsuk Chung 和 Soyeon Caren Han。2023年。Form-NLU：表单语言理解数据集。载于第*46*届国际 *ACM SIGIR* 信息检索研究与发展会议（*SIGIR '23*）。ACM，美国纽约州纽约市，10页。https://doi.org/XXX

允许免费制作本作品全部或部分内容的数字或纸质副本，供个人或课堂使用，但副本不得用于或分发以获取利润或商业利益，且副本必须包含本声明和首页上的完整引文。本作品中归其他方所有的组件，其版权必须得到尊重。注明出处后可以摘录。若要以其他方式复制、重新发布、发布到服务器或重新分发至列表，则须事先获得特定许可和/或支付费用。请向 permissions@acm.org 申请许可。*SIGIR '23*，*2023*年*7*月*23*日至*27*日，台北，台湾© 2023年 美国计算机协会。ACM ISBN 978-1-4503-XXXX-X/18/06...$15.00 https://doi.org/XXX

1 引言

文档中的结构信息及其值的提取可以成为自然语言处理（NLP）任务的重要信息来源，尤其是信息抽取和检索。近年来，NLP 社区和 IBM、Microsoft 等行业企业提出了一系列技术，以理解 文档的空间和/或逻辑结构[14, 17, 29, 32], 并提取 关键信息[4]。这些研究主要集中于基于视觉丰富文档（VRD）的任务，例如学术论文[17, 32]、收据[9, 16], 和表单[10, 27], ，并解决了许多基准数据集问题，包括版面分析[13, 32]、表格结构识别 [2, 30, 31], 和文档问答 [15, 21]。

大多数基于 VRD 的问题都已得到成功解决，然而，表单理解任务仍颇具挑战性。这主要有两个原因：表单中存在两类作者，以及多种视觉线索相结合。首先，与通用 VRD 不同，表单的主要目的非常明确：从表单用户处收集数据；这一目的适用于从医疗领域到行政数据收集的各种场景。基于这一目的，我们可以认为表单主要有两类作者：表单设计者和表单用户。表单设计者专注于构建表单结构，通过定义明确的关键信息点来收集所需信息。开发完成的表单会用作用户界面，让表单用户根据自己的理解填写表单值。遗憾的是，并非每种表单都清晰易懂。为了收集多种所需信息，一些表单设计者倾向于制作具有不同布局/结构的表单，而这些表单的语义实体之间存在复杂的逻辑关系和位置关系。表单用户很容易对设计者的意图感到困惑，进而导致键值对的错误对齐。对表单设计者意图的困惑以及表单用户的不确定性，会增加表单文档理解和信息抽取的难度。其次，由于涉及表单开发者与用户之间的关系，表单很可能兼具不同形式，例如数字、印刷体或手写体。例如，设计者可能向用户提供电子版纸质表单，而用户则可能通过多种载体提交填写后的表单，例如数字版、印刷版或手写版。这种情况下，用户提交的表单中也常会出现各种噪声

（分辨率低、扫描不均匀或字迹潦草）。这给理解表单文档结构和提取关键键值对带来了巨大困难。近年来，已有多个表单理解数据集发布，数据收集自扫描收据、 [9, 16], 合同[19], 和跨领域表单 [10, 27]（见表格1）。然而，这些数据集将表单开发者的意图设定得相对简单、通用，未能应对对表单设计者意图的困惑以及表单用户的不确定性。此外，大多数数据集未涵盖不同载体的文档版本及其噪声。这会加大理解表单结构和提取关键信息的难度。

本文介绍了一个用于表单结构理解和关键信息提取的新数据集。该数据集有助于解读表单设计者的特定意图，并将用户填写的值与之对齐。我们的数据集还涵盖三种表单类型：数字、印刷体和手写体，囊括多种表单外观和布局，并处理其中的噪声。此外，我们提出了一个用于表单结构理解和键值信息抽取的新基线模型，该模型利用稳健的位置关系和逻辑关系。为此，我们涵盖由用户导致的表单多样性，以便从表单中准确提取关键信息；这正是当前新兴的工业需求。我们的模型涵盖文档的层次结构，从词语到句子、从句子到段落等语义实体，再到文档页。请注意，现有的基于 Transformer 的模型主要独立关注词元 [5, 8, 25, 26, 28] 或实体级别[12, 20, 29]，忽视了不同层级元素之间的上下文依赖。在评估中，我们首先考察用于表单布局理解的强大目标检测模型，然后测试所提出的关键信息提取任务模型。此外，我们还使用现成的 PDF 布局提取工具评估了关键信息提取数据集和所提出的模型，并验证了其在真实场景中的可行性。

本研究的主要贡献可总结如下：1）我们引入了 Form-NLU，这是一种新的表单结构理解和关键信息提取数据集，涵盖了特定的表单设计者意图，并实现了与用户值的对齐。2）我们提出了一种新的基线模型，用于处理文档中的位置关系、逻辑关系和层次结构。该模型在 Form-NLU 上的表现优于其他最先进水平的表单关键信息提取模型。3）我们将所提出的数据集和模型与现成的 PDF 布局提取工具相结合，并证明了其在真实世界表单文档场景中的可行性。

2 相关工作

已有研究介绍了通用文档理解基准数据集 [17, 21, 32]，但我们在此简要介绍表单文档理解与信息抽取。它涉及多方交互，因此会产生更复杂的位置关系和逻辑关系，如表1所示。我们想讨论并与以往基准数据集比较三个主要方面。首先，大多数基准数据集 [10, 24, 27] 未涵盖文档版本的各种载体及其噪声。例如，表单设计者可能提供数字表单，而表单用户可能通过各种载体提交填写后的表单，例如数字

Name	来源	Type			特征设计者		
		$\mathcal{D}$	$\mathcal{P}$	$\mathcal{H}$	B.Box	文本意图	
FUNSD [10]	噪声表单	×	○	○	○	○	通用
XFUND [27]	合成表单	○	×	○	○	○	通用
EPHOIE [24]	试卷	×	×	○	○	○	通用
Charity [19]	年度报告	◐			×	○	特定
NDA [19]	协议	○	×	×	×	○	特定
Form-NLU（**我们**）	金融表格	◐			○	○	特定

表1：表单理解数据集概述。

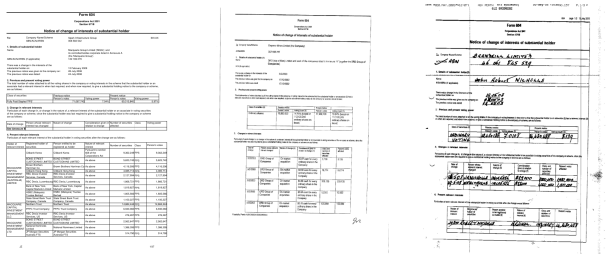
（D）、印刷体（P）或手写体（H）版本。这种情况常常会因各种噪声（包括分辨率或扫描问题）而影响表单结构理解和信息抽取的质量。因此，成功的基准数据集应涵盖具有各种载体的真实场景。其次，为了进行表单结构理解和信息抽取，解读表单组件之间的位置关系和逻辑关系至关重要。大多数基准数据集通过使用边界框（B.Box）和文本信息来支持对位置关系和逻辑／语义关系的理解。然而，Stanisławek 等人 [19] 发布了两个表单信息抽取数据集——NDA 和 Charity，却未提供语义实体的边界框坐标。最后，表单文档有两类主要作者：表单设计者和表单用户。表单设计者设计表单结构以收集所需信息（设计者意图），而表单用户则试图理解设计者意图，却很容易感到困惑。因此，处理设计者意图对于理解表单结构和提取关键信息至关重要。一些流行的基准数据集 [10, 27] 将表单开发者意图设定得相对简单且通用，未能处理表单设计者意图所带来的困惑以及表单用户的不确定性。例如，这些数据集涵盖简单的表单组件类型、键和值。扫描试卷 [24] 数据集的布局相对简单，采用水平键值对结构，不包含任何动态布局组件，例如表格、段落或复杂的键值对。**Form-NLU** 是首个视觉语言表单理解数据集，旨在支持研究人员解读特定的设计者意图，并应对用户输入所带来的、源自各种表单载体的噪声。

3 数据集

3.1 数据收集

我们的 **Form-NLU** 是 SIRCA 收集的 604号表格（重要股东持股权益变动通知）金融表单数据源的一个子集[1]。它提供了 2003年 至 2015年提交给澳大利亚证券交易所（ASX）的每份重要股东通知表的文本记录[2]。为了更好地理解表单设计者的意图，我们纳入了十二个最重要的表单字段 [1]。每个表单字段都要求用户提供一个**值**，其中应包含相应**键** 所表达的表单设计者意图所询问的特定信息。例如，“公司名称” 这个键预期对应一个给出公司名称的字符串值，而“投票比例” 则要求

[1]https://asic.gov.au/regulatory-resources/forms/forms-folder/604-notice-of-change-of-interests-of-substantial-holder/[2]https://www2.asx.com.au/



(a) 数字 ($\mathcal{D}$) **(b) 印刷体（$\mathcal{P}$）** **(c) 手写体（$\mathcal{H}$）**

图 1：数字、印刷体和手写体表单样本。

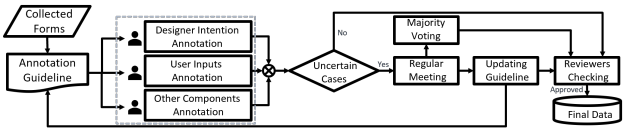


图2：总体标注工作流。

提供一个表示投票所占比例的数字。为了涵盖现实场景中前述开发者与用户之间的关系，我们还纳入了三种格式变体，具体取决于不同用户填写和提交表单的方式：(1) 数字（D）、(2) 印刷体（P）和 (3) 手写体（H）。数字表单以电子方式填写，并直接以 PDF文件形式提交。印刷体和手写体表单分别以电子方式和手写方式填写，然后先进行扫描，再以 PDF文件形式保存并提交。图 1 展示了每种表单类型的一个示例，体现了表单格式可能存在的多样性，例如字体样式和旋转方向。

3.2 标注指南

根据我们的任务目标（见第5节），我们首先制定了标注模式和指南。对于布局分析（即任务A[3]），我们创建了七种不同的表单布局组件类别（标题、章节、表单键、表单值、表格键、表格值 和其他），涵盖了所收集表单中的所有可能组件。每个表单布局组件都代表文本区域中的一个语义片段。标题 和 章节 分别对应表单文档的标题和各个章节。表单/表格键和表单/表格值 分别指表单设计者拟定的问题（例如，公司名称）及对应的用户输入字段。表单中的键/值 与表格中的键/值之间的区别在于，根据我们的定义，表单键值 对是水平对齐的，而表格键值 对是垂直对齐的。不属于这四种类型的组件均定义为其他，例如自定义段落，以及一些公司出于自身目的添加和修改的单元格。

而布局分析（作为任务A）侧重于通过这些通用语义组件理解整体表单结构，关键信息提取（作为任务B[4]）则旨在理解表单设计者的不同意图。因此，在任务B中，任务A中的键和值组件会根据

[3]参见第5.1节 任务A——表单布局分析 [4]参见第5.2节 任务 B—— 关键信息提取

十二种键意图进一步区分，包括：(1) 公司名称 (com_nm)，(2) 公司ID (com_id)，(3) 持有人名称 (hold_nm)，(4) 持有人ID (hold_id)，(5) 变更日期 (chg_date)，(6) 上次通知日期 (gvn_date)，(7) 上次通知日期 (ntc_date)，(8) 证券类别 (class)，(9) 先前股份 (pre_shr)，(10) 先前投票百分比 (pre_pct)，(11) 股份 (new_shr)，(12) 新投票百分比 (new_pct)。键(1)-(2)和键(3)-(4)分别询问上市公司和主要持有人的身份，而键(5)-(12)则查询发生变更的股份值。我们只需在十二个键名称后分别附加“键”和“值”标识符，相应地为这些键值对创建二十四个标签。例如，每个键值对会生成两个标签，如“公司名称键”和“公司名称值”。基于这些不同的键意图，我们检查了值模式，以探究用户理解多样化和不确定性可能造成的噪声。统计分析见第4章（图3）。

3.3 标注流程

我们招募了三名人工标注员和两名人工审核员[5]，并使用DataTorch[6]对每个表单组件进行迭代式标注。具体而言，如图2所示，我们将标注分为三个专门的子任务：(1) 对上述12种类型的键进行标注，(2) 对配对的值进行标注，以及(3) 对其余非键、非值组件进行非键值标注，包括标题、章节和其他。每位标注员分别负责一个子任务。他们识别所有可能的目标组件，并根据标注指南标注相应的矩形边界框和标签。标注过程中，标注员会标记任何不确定的情况，并在例会中通过讨论和多数表决进一步确定。必要时，我们会更新标注指南，以便之后处理类似情况。例如，一种典型的不确定情况是表单键的标注。由于不同表单用户采用的格式各异，例如“上次通知日期为 *2003*年*2*月*12*日”这一关键信息（7）“上次通知日期”的示例，最终商定的键“上次通知日期为”是一个完整的句子，其对应的值为“*2023*年*2*月*12*日”。这一迭代标注流程最终产出了757份已标注的数字表单文档。此外，我们还随机选取了50份印刷体表单和50份手写表单，并按照与数字表单相同的流程进行标注。我们将这些印刷体和手写表单作为额外的测试集，以探索在现实场景中处理不同表单类型的可能性。

为确保人工标注的最终质量，我们将所有表单的已标注边界框和标签可视化，并分配给两位人工审核员进行并行人工检查。他们参照最新的标注指南逐一检查每份表单，并标注其正确性。对于两位审核员都标注为错误，或两人意见不一致（即一人标注为正确而另一人标注为错误）的表单，他们会进行复核。随后，

[5]三位标注员均具有悉尼大学计算机科学或金融领域的背景，而两位人工审核员则是金融领域专家。[6]https://datatorch.io/features/annotator

根据两位审核员最终达成一致的决定对这些表单进行修改。为衡量标注质量，我们使用 Cohen's Kappa [3] 和汉明损失计算标注一致率，得出的总体分数分别为**0.998** 和**0.003**，表明标注质量很高。

3.4 标注格式

我们提出的 **Form-NLU** 的最终标注以 *.json* 格式提供，分别对应任务A和任务B[7]。每个表单片段的标注都存储在字典对象 中，该对象包含多个键值对，键名具有指示性，位于 *.json* 文件中。两个任务中每个表单片段共有的主要属性包括：表示边界框坐标的边界框 、表示该特定片段文本词元的文本 （例如来自光学字符识别或 PDFMiner[8]），以及根据具体任务为该片段指定的标签 ，例如任务A中片段的"章节"，或任务B中片段的"公司*ID*（*com_id*）"。此外，我们还纳入了一些基于实验得出的辅助属性，以供未来开发使用，例如任务A的分割 坐标，以及任务B的视觉_特征 和***BERT_CLS*** （参见第6.4、6.5节）。

4 数据集分析

4.1 组件分布

我们标注的 **Form-NLU** 最终版本包含857份表单，其中包括757份数字表单、50份印刷体表单和50份手写体表单。表2展示了数据集划分的总体统计信息，以及表单组件的分布明细。我们按70/10/20的比例随机划分数字表单，得到535份训练集表单、76份验证集表单和146份测试集表单。此外，50份印刷体表单和50份手写体表单作为额外的测试集划分。分布明细显示，键 和值 持续占据主导地位，因为它们是表单的主要内容。由于存在值为空的情况，值 的数量可能少于键 。标题 和章节 的出现频率最低，且相近，这是因为每份表单文档通常包含一个主标题和可选的一到两个副标题，同时始终包含两个必要章节。其他 的出现频率略高于标题 或章节，具体频率取决于不同的表单填写者。表单整体分布如此一致，归因于使用了604号表格模板。这使我们的**Form-NLU** 成为一个有潜力的基准数据集，可用于探索在标准表单模板可用时解决真实世界金融表单布局理解问题的方法（我们的任务A）。此外，我们进一步区分键和值，以理解特定的表单意图（我们的任务B）。这些键值对位于表格 （多为纵向键值对对齐）或表单 （多为横向键值对对齐）中，这表明空间关系多种多样，模型需要学习这些关系，才能准确识别值并将其与相应的键对齐。表3列出了每种表单组件的平均边界框尺寸和文本词元数量，反映了其视觉和语言特征，这些特征可能有助于

[7]我们提供了两个任务中若干真实的*.json* 文件示例，详见 https://github.com/adlnlp/form_nlu#dataset-loading-and-samples。本文一经录用，我们将公开完整数据集。[8]我们使用 PDFMiner 提取数字原生表单的文本$\mathcal{D}$，并使用 Google Cloud Vision 提取印刷体和手写体数据集的文本（$\mathcal{P}$,$\mathcal{H}$）

Type	用法	Form 图像	标题部分		Form 键值		表格 键值		其他
训练	数字	535	1068	1070	3708	3568	2669	2669	1691
Val	数字	76	152	152	524	510	380	379	246
Test	数字	146	292	292	1009	978	730	730	458
	印刷体	50	98	100	346	332	250	249	152
	手写体	50	100	100	348	315	249	226	149
总数		857	1710	1714	5935	5703	4278	4253	2696

表 2：每个数据划分的表单组件数量。

Type	标题	章节	Form Key	Form 值	表格 Key	表格 值
边界框平均宽度	186.15	136.82	95.46	85.98	57.29	41.92
边界框平均高度 30.44 12.47			12.47	12.47	10.50	10.45
边界框平均 PX 4275.94 1720.62			1204.96	1183.10	609.75	456.94
平均词元数	7.35	7.10	5.16	4.14	4.29	1.74

表3：平均边界框宽度、高度、数量的每个表单组件的像素（PX）和词元数量

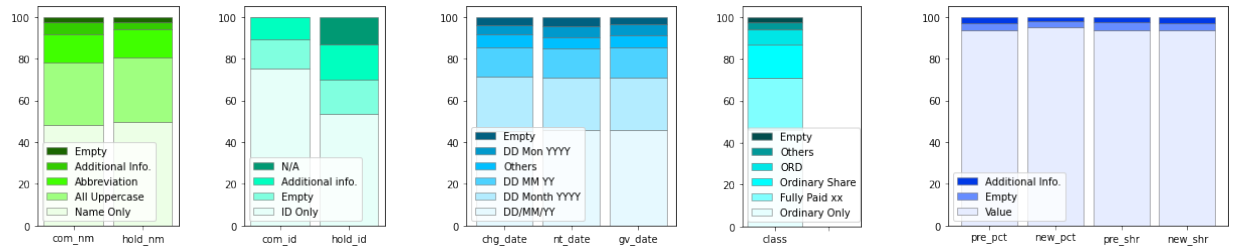
特定键和值的识别。标题 和章节 通常包含较长的文本，组件尺寸也较大，字体也更大。相比之下，键 和值 都短得多，且位于较小的边界框内。尤其是表格 中的值 长度最短，因为它们大多是数值。

4.2 值模式比例

如前所述，由于表单用户的理解存在差异和不确定性，填写内容也各不相同，并且可能包含噪声。为了展示填写内容的多样性，我们总结了每种键意图对应的主要值模式，并在图3中给出了各模式所占的比例。在所有键中，占比最高的值遵循一种常见的值模式，即表单用户倾向于仅提供满足键意图所需的最少信息。例如，在图3a中，他们只需在键 *com_nm* 中填写公司名称，或在图3b中仅填写标识符。然而，还存在其他几种值模式，包括图3a、3b和3e所示的附加信息，以及采用不同的数据格式或书写风格，例如图3c中出现的各种日期格式。这些由人为因素造成的变体充分反映了现实世界中的表单理解场景，也揭示了我们在任务B中实现精准信息检索所面临的潜在挑战。总体而言，股份类别（图3d）和日期类型值（图3c）呈现出相对更多的模式，而图3e中的数值类型值则倾向于与常见模式保持更高的一致性。具有相同数据类型值的不同键之间存在这种模式相似性，这表明，仅依赖语言线索可能并不足够；为了区分不同键对应的这些相似值，还需要理解表单中各个值的相对空间位置。

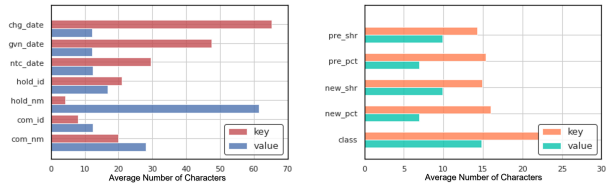
4.3 键值对比较分析

为了深入分析键值对的性质，我们进一步分别总结了图4和图5中十二个键值对的平均字符数和空间关系比例。可以观察到，标准的604表单模板通常包含七个固定的表单键值对（即图4a/5a）和另外五个固定的表格键值对（即图4b/5b）。这两组键值对各自展现出不同的特征。如图4a所示，对于基于表单的键值对，



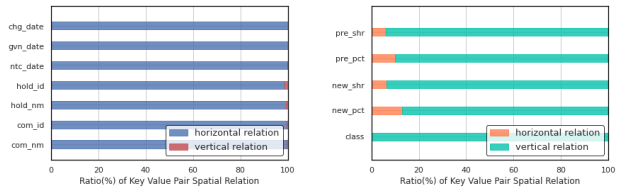
(a) 名称 (b) 标识符 (c) 日期 (d) 类别 (e) 百分比/份额

图 3：各键组的值模式分布。



(a) 表单键值对 (b) 表格键值对

图 4：每个键值对中的平均字符数



(a) 表单键值对 (b) 表格键值对

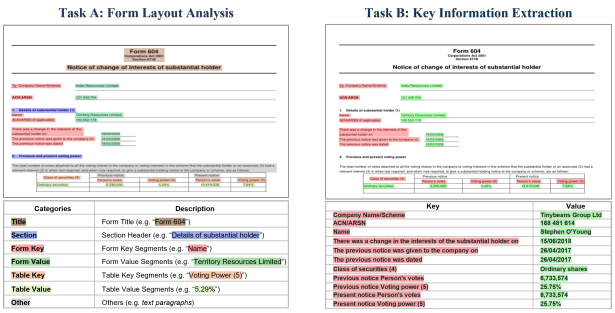
图 5：各键值对的水平和垂直关系比例（%）

三个日期类型键值对的值通常比键长得多，而其他四个键值对的值则可能比键长，也可能比键短。*hold_nm* 键值对的字符长度差最大。相比之下，图4b中的五个基于表格的键值对，其值都比键短，且长度差异相近。总体而言，大多数键对应的值都很短（例如，＜20 个字符），因此可提供的语义上下文可能有限。另一方面，如图5所示，七个基于表单的键值对都水平对齐，而表格中的另外五个键值对则大多为垂直排列。不过，两组中也都观察到一些对齐方向相反的特殊情况，例如基于表单的组中*hold_id*/*hold_nm* 的垂直键值对（图5a），以及*com_id*/*com_nm* 的垂直键值对（图5a）；此外，基于表格的组中，四个百分比类型的键和值呈水平排列（图5b）。因此，仅仅记住每个键的空间对齐方式，无法在任务B中实现最优的值检索。

5 任务概述

5.1 任务A——表单布局分析

任务A的目的是检测表单中的语义实体（标题、章节、表单_键、表单_值、表格_键、表格_值、其他）。



(a) 任务A示例 (b) 任务B示例

图6：任务A和任务B的示例。对于任务A，用户需要检测每个布局组件的边界框并识别相应类别。任务B要求用户将固定键文本（红色区域）输入模型，以预测感兴趣区域索引对应的值（绿色区域）

这是理解表单布局的前置任务，方法是检测每个语义实体的位置，包括键、值和其他对象。给定一张表单图像 I，使用目标检测模型检测一组感兴趣区域（r_1、r_2、...、r_n）。每个 r_i都包含边界框 b_i坐标（x_i、y_i、w_i、h_i）及其语义类别 c_i。如图 6a 所示，输入表单图像中每个语义实体的边界框都会被检测出来，并根据识别出的类别着色。模型应能有效检测表单布局组件，例如表单标题（如“604号表格”）和章节标题（如“主要持有人的详细信息”）。[illegible]我们还希望模型能够区分表单或表格中的键/值，例如，“名称”应被检测为表单_键实例，而“投票权（5）”应被检测为表格_键实例。

5.2 任务B -关键信息提取

任务B旨在评估所提出的模型能否全面理解设计者意图和用户不确定性，从输入表单中提取有价值的信息。训练和推理阶段均可使用真实感兴趣区域集合 $R^{gt} = (R_1, R_2, ..., R_n)$。给定键文本信息 t、文档图像 I以及一组真实感兴趣区域 $(r_1, r_2, ..., r_n)$，模型 H可以输出与输入 t对应的感兴趣区域索引编号。如图6b所示，每个以红色背景突出显示的实体都是我们需要提供给所提出模型的键 t。

，绿色配对感兴趣区域的索引是模型根据当前输入键给出的期望输出。例如，输入所需的键文本内容“ 公司名称/计划”，模型的期望输出是与之配对的值所在的感兴趣区域索引，我们可[redacted]（“ Tinybeans Group Ltd”）。

6 评估设置

6.1 实现细节

对于任务A，我们分别基于Detectron2平台，在我们的数据集上对采用两种骨干网络[9]的Faster-RCNN和Mask-RCNN模型进行微调。我们将5000、128和0.02分别设定为最大迭代次数、批量大小和基础学习率，其他设置均与Detectron2官方教程[10]相同。对于任务B，我们采用多种方法对视觉与语言特征进行编码。首先，所有任务B采用的基线模型均使用预训练的BERT对关键文本内容进行编码。此外，在视觉方面，VisualBERT、LXMERT和M4C模型均使用从ResNet101的Res5层提取的2048维特征。输入关键文本词元的最大数量以及每页的分段数量分别设定为50和41。任务A和任务B的实验均在配备CUDA 11.2的51 GB Tesla V100-SXM2上进行。

6.2 任务A基线模型与指标

我们使用两个流行的目标检测模型——**FasterRCNN** [18]和**MaskRCNN** [6]，搭配不同的ImageNet预训练骨干网络，在我们的数据集上进行测试。为评估目标检测模型的性能，我们采用 *mAP*（平均精度），这是目标检测任务中常用的指标。 *mAP*分数通过对所有类别在预定义交并比阈值[11]上的 *APs* 进行平均计算得出。

6.3 任务B基线模型与指标

任务B主要关注根据输入文本内容预测RoI索引对应的值（如图6b所示）。因此，我们在**Form-NLU**上采用了若干多模态Transformer框架作为基线模型，其输入是多方面的RoI特征，包括大型预训练**VisualBERT** [12], **LXMERT** [20] 以及未预训练的**M4C** [7] 模型[12]。在评估方面，我们使用加权F1分数作为主要评估指标，以表示总体和细分性能。需要注意的是，某些视觉语言预训练模型（例如ViLT [11], LayoutLM [25],）被排除在外，因为它们主要基于图像图块或片段，不符合该任务的需求。

6.4 任务A模型

对于任务A——表单布局分析任务（章节 7.1），我们采用 Faster-RCNN 和 Mask-RCNN，并使用不同深度的骨干网络（ ResNet-50 和 ResNet-101）作为基线模型，开展对照实验，以检验不同模型架构和模型规模的影响。此外，

[9]Faster-RCNN 骨干网络包括 faster_rcnn_R_50_FPN_3x、faster_rcnn_R_101_FPN_3x, Mask-RCNN 则包括 mask_rcnn_R_50_FPN_3x、mask_rcnn_R_101_FPN_3x [10] https://colab.research.google.com/drive/ [redacted] 5 jcaJoc bCFAQ jDe [11] HwtXj BMD -m 我们参考 [32] 采用 *mAP*作为指标，并沿用其针对任务A设定的阈值。[12]详细的基线设置可见 https://github.com/adlnlp/form_nlu#baseline- model-description

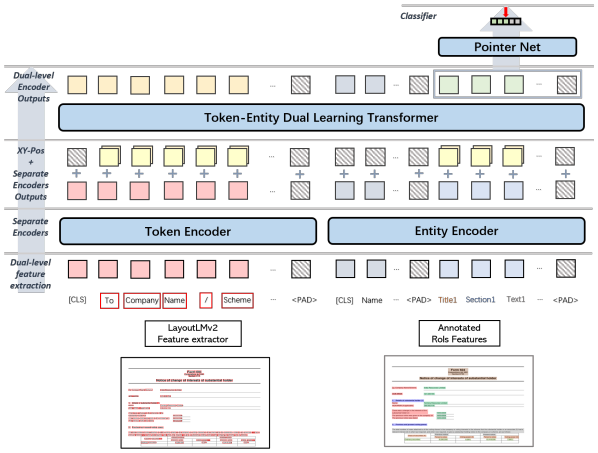


图7：我们为任务B提出的关键信息提取模型架构

对于真实世界案例分析（章节 7.3），我们采用了一些使用开源 PDF 解析器（如 PDFMiner）的方法，在不训练深度学习模型的情况下分析表单布局。PDF 解析器的输出也可作为任务B的输入。

6.5 任务B模型

对于任务B，我们提出了一种新的文档关键信息提取模型（如图7所示），用于根据输入的键内容预测相应的值。为此，我们通过两个组件研究位置和逻辑关系及其层次结构：多方面特征提取与整合，以及实体-词元双层模型。

6.5.1 多方面特征。 我们研究了五种方面特征，包括视觉特征（V）、文本特征（T）、位置特征（P）、密度特征（D）和间距特征（G），并采用相应的编码方法，全面探究哪些特征有助于理解设计者意图，以及论证类表单文档片段表示。许多传统的文档理解任务方法仅展现了视觉、文本和位置特征的有效性与重要性 [25, 28, 29]。除了这三种方面特征，我们还旨在探究文本密度对表单理解的有效性； [14]已证明文本密度有助于文档布局分析。此外，根据数据集分析结果，实体之间的间距对于理解表单结构至关重要。因此，我们引入了两种布局相关特征，包括归一化位置特征（边界框坐标）以及实体与其邻近实体之间的间距 [13]。

6.5.2 实体-词元双层模型。 我们提出的新型实体-词元双层表单信息抽取模型包含四个组件，包括实体编码器 $Encoder_{entity}$、词元编码器 $Encoder_{token}$、双编码器 $Encoder_{dual}$ 以及基于 Pointer Net 的分类器 ϕ。

[13]我们的 Form-NLU Github（https://github.com/adlnlp/form_nlu#multi-aspect-features）提供了更详细的特征表示方法。

1) 实体编码器 旨在学习实体之间的语义关系，以增强基础实体表示。根据初步结果，我们选用预训练的 LXMERT（cite tan2019lxmert）作为实体编码器。基于 LXMERT 的实体编码器输入包括键文本 T_{key}、实体的视觉特征 V_{entity}以及边界框坐标 B_{entity}。将这些特征输入预训练的 LXMERT 后，即可提取增强后的实体感兴趣区域表示 $E \in \mathbb{R}^{768}$ 。

2) 词元编码器 旨在获取跨模态词元表示。我们采用最先进的布局感知预训练视觉语言 Transformer LayoutLMv2 [28], 作为词元编码器。我们首先使用*LayoutLMV2* 特征提取器 提取并编码词元级特征，包括词元文本 T_{token}、词元感兴趣区域的边界框 B_{token}及其像素特征 I_{pixel}。然后，我们遵循原始 LayoutLMV2 的设置来处理提取的词元级特征，并将其输入预训练的 LayoutLMv2 词元级编码器$Encoder_{token}$，以获得词元级表示 $T \in \mathbb{R}^{768}$。

3) XY-Pos 双编码器旨在学习对几何形状敏感的多粒度、多模态特征表示。从预训练模型中获取实体级和词元级特征表示后，我们将其作为双层互学习编码器 $Encoder_{dual}$ 的序列输入，其中采用具有 8 个头 自注意力的 6 层 Transformer 作为基本框架。不同于采用正弦位置编码 [22], 的原始 Transformer，我们提出了一种新的几何敏感位置编码 *XY-Pos*。它沿 X 轴或 Y 轴展平堆叠的归一化边界框坐标集合，以获得X_i^{pos}和 Y_i^{pos}。假设 $b_i = (x_i,\ y_i,\ w_i,\ h_i)$ 是文档页 D_j 中 $Encoder_{dual}$的 r_i的边界框，则 D_j的大小为（$W_j,\ H_j$）。

$$pos_i^x = [\frac{x_i + d_{i_1}}{W_j}, \frac{x_i + d_{i_2}}{W_j}, ..., \frac{x_i + d_{i_m}}{W_j}] \tag{1}$$

其中第 l 步 $d_{i_l} = \frac{w_i \times l}{m}$ ——

$$X_i^{pos} = flatten([pos_i^x] \times n) \tag{2}$$

我们将 m 和 n 设为 32 和 24，从而得到一个 768 维向量。最后，将 X_i^{pos} 展平为一个 $m \times n$ 维向量，得到 $X_i^{pos} \in \mathbb{R}^{768}$。我们采用类似的步骤生成沿着 r_i 的Y 轴的位置编码 Y_i^{pos}。馈入 $Encoder_{dual}$ 之前的最终输入表示可表示为:

$$E_{pos_{xy}} = E + X_{entity}^{pos} + Y_{entity}^{pos} \tag{3}$$

$$T_{pos_{xy}} = T + X_{token}^{pos} + Y_{token}^{pos} \tag{4}$$

与大多数视觉语言预训练模型不同， [20, 28] 采用位置编码方法，通过线性投影将 4 维边界框坐标映射为高维向量。*XY* 位置编码能够捕获实体与词元之间更多的几何特征。接着，我们将它们馈入 $Encoder_{dual}$，并得到更新后的词元和实体表示 T_{dual}、 E_{dual}。

4) 基于 Pointer Net 的分类器 将多方面特征与 E_{dual}拼接，得到 E_{multi}。将 E_{multi}输入基于 Pointer Net 的分类器 [23]，得到一个分数向量，该向量经过 $softmax$后用于获取最终预测结果 y_{pre}。

测试集	模型	*mAP*	细分项（*Precision*）							
			标题部分	表单键		表单值	表格键		表格值	其他
$\mathcal{D}$	**F-50**	68.98	73.54	69.36	67.93	70.37 <style 标识符="1">69.18			66.76	65.73
	F-101	69.99	71.32	68.30	72.37	71.70	68.74		72.84	64.65
	M-50	71.74	77.29	69.06	**73.01**	70.30	68.40		73.25	**70.90**
	M-101	**71.93**	**79.54**	**69.64**	71.25	**72.38**	67.02		**74.38**	69.33
$\mathcal{P}$	**F-50**	56.46	57.44	53.98	54.49	51.46	59.31		62.86	55.67
	F-101	59.54	55.72	52.81	63.13	61.32	59.42		68.11	56.26
	M-50	63.06	59.83	60.98	63.74	**61.91**	70.75		65.93	58.29
	M-101	**65.47**	**63.05**	**62.49**	**66.84**	61.80	**71.82**		**70.19**	**62.10**
$\mathcal{H}$	**F-50**	49.87	58.73	43.85	63.14	25.15	63.48		40.14	54.56
	F-101	50.39	53.92	33.36	65.28	36.68	66.97		41.06	55.43
	M-50	57.99	61.45	48.46	68.16	43.14	71.76		**56.95**	56.05
	M-101	**60.22**	**64.81**	**54.46**	**69.97**	**46.82**	**73.34**		52.81	**59.32**

表格 4：版面分析任务（任务A）的总体性能及分项结果。F-50 和 F-100 分别表示以 ResNet50 和 ResNet101 为骨干网络的 Faster-RCNN。Mask-RCNN 也呈现相同规律（M-50 或 M-101）。

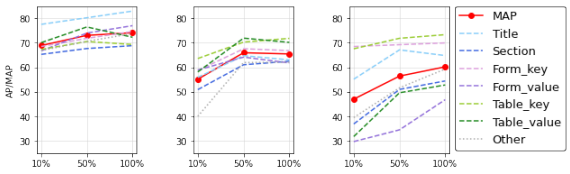
7 结果

7.1 任务A：表单布局分析

7.1.1 总体性能与细分性能。本节展示版面分析（任务A）模型在数字（D）、印刷体（P）和手写体（H）数据集上的测试性能。从表格 4 可以观察到， *mAP* 的 Mask-RCNN 相较于采用相同骨干网络的 Faster-RCNN，在 D、P 和 H 数据集上的表现分别高出约 2%、6% 和 10%。这可能是因为 Mask-RCNN 采用的辅助实例分割任务能够实现更精细的空间定位。接着，我们探究不同骨干网络的影响，例如采用 ResNet-101（F-101）的 Faster-RCNN，或采用 ResNet-50（M-50）的 Mask-RCNN。从表格 4 可以发现，F-101 和 M-101 的表现始终比 F-50 和 M-50 高出约 1% 至 2%。这表明，更深或更大规模的预训练骨干网络可能生成更全面的视觉表征。值得注意的是，D 的总体性能比 P 高出约 5%，甚至比 H 高出约 11%。主要原因可能是扫描表单（包括印刷体 P 和手写体 H）与数字表单之间存在明显差异，尤其是手写表单涉及更多用户不确定因素，例如书写错误或扫描旋转。

由于我们的 Form-NLU 提供了细粒度的布局组件类型，例如将 值 细分为 表单_值 和 表格_值，因此我们能够观察并分析不同布局位置分布对细分组件性能的影响。例如，某些布局组件在三个测试集中的表现稳定，例如 表单_键 和 表格_键。这可能是因为这些组件由表单设计者设计，具有更多共通的布局和视觉模式，例如字体和布局排列。然而，表单_值 和 表格_值 组件可能包含更多来自用户、电子设备或传播过程的不确定性和噪声，导致其在 H 上的 *mAP*（46.82% 和 52.81%）低于 P（61.80% 和 70.19%），且远低于 D（72.38% 和 74.38 %）。此外，对于 标题 和 其他 组件，M-101 在 D 上的表现（79.54% 和 69.33%）明显优于 P（63.05% 和 62.10%）和 H（64.81% 和 59.32），而 P 和 H 的表现几乎相同。这可能是因为扫描表单（P 和 H）与数字表单之间存在差异，例如扫描表单的旋转或较低的分辨率。

7.1.2 阶梯式训练集比例。 我们设置了不同的训练集规模（T）比例（10%、50% 和 100%）来训练 M-101，并在 D、P 和 H 集上对其进行评估，以探究训练规模对任务A的影响。从图8可以看出，随着 T 规模增大，总体性能和



(a) 数字(<style 标识符='1'>数字) (b) 印刷体($\mathcal{P}$) (c) 手写体($\mathcal{H}$)

图8：任务A中数字（$\mathcal{D}$）、印刷体（$\mathcal{P}$）和手写体（$\mathcal{H}$）测试集上的不同训练比例 *mAP*

细分性能均有所提升；不过，当 T 达到 50% 时，这一趋势在 D 和 P 集上变得不明显。就 H 集上的性能而言，扩大训练规模可以提升所有类型版面组件的性能，其中表格_值、部分 和其他的提升尤为显著。特别是，作为下游任务的一种重要版面组件类型，表格_值 在 H 集上约占 30%。总体而言，图8所示的趋势表明，我们的训练集（T）包含多种表单类型，因此即使 T 的比例较小，也能够应对不同的应用场景。此外，这也表明，扩大数字训练集（T）的规模能够有效提升印刷体和手写体的性能。增加训练样本可能有助于捕捉更多共享特征模式，例如特定组件的空间分布、排版相似性、字体类型和字号。

7.2 任务B：关键信息提取

7.2.1 总体性能与多方面特征性能。 本章节比较并分析了我们的模型与三种广泛使用的基线模型在不同配置下，于数字化（D）、印刷体（P）和手写体（H）测试集上的性能。首先，我们重点比较原始模型之间的性能（表5中每组模型的第一行），这些模型的输入特征和模型架构均与其原始论文中的描述保持一致。我们的模型在D、P和H测试集上的表现均优于所有原始基线模型。这表明，利用词元级特征进行迭代学习能够增强实体级表示，从而提升下游任务的性能。此外，在三种原始基线模型中，LXMERT在D、P和H测试集上的表现分别比 VisualBERT高出2.28%、6.38%和15.36%。这可能是因为 LXMERT的输入包含位置信息，而VisualBERT仅基于输入感兴趣区域的视觉特征进行预训练。随后，与VisualBERT相比，未经预训练的M4C在D、P和H测试集上的性能分别提高了约 1.5%、3%和9%。这可能说明文本特征和位置特征具有重要意义。然而，由于不同原始模型之间的输入特征存在差异，为了进行公平比较，我们开展了外部实验，以探究多方面特征对所有所采用模型的影响。

M4C 最初包含 V、T和 P个输入感兴趣区域，并且是一个使用随机初始权重的非预训练模型。因此，在三个测试集上向模型中添加 D和 G后，表5中只能看到轻微的提升。对于 VisualBERT，原始模型只包含感兴趣区域的视觉特征；叠加更多方面特征后，性能逐步提升，其中 $\mathcal{H}$上的 F1分数从 65.25%（原始模型）提升至 72.65%（添加 V、T、G后）。添加 D和 G到 VisualBERT 后，性能明显提升，这可能是因为额外特征使输入

模型	输入特征					总体（F1分数）		
	V	T	P	D	G	数字(D)	印刷体(P)	手写体(H)
M4C	○			×	×	96.91	88.62	74.06
	○				×	96.32	89.52	74.98
	○			×	○	97.00	88.81	75.04
	○					97.22	89.78	74.89
VisualBERT	○	×	×	×	×	95.55	85.91	65.25
	○	×	○	×	×	95.84	85.93	67.25
	○			×	×	96.07	85.90	70.14
	○			×	○	96.61	87.43	70.79
	○				×	96.70	87.00	71.28
	○					96.73	87.18	72.65
LXMERT	○	×	○				92.29	80.51
	○			×	×	97.67	94.15	82.80
	○	×	97.70				94.59	82.60
	○			×	○	97.74	94.49	83.71
	○			○	97.		95.07	84.43
我们的模型	○						95.07	84.56
	○				×	**99.30**	95.32	84.75
	○	○	○	×	○	99.12	95.50	85.77
	○				○	99.	**95.50**	**86.98**

表5：不同背景颜色。这用于展示 原始模型（每组模型的第一行）、包含 的模型（每组模型的最后一行）在 兴趣区域输入包含视觉（V）、文本（T）、位置（P）、文本密度（D）、间距（G）特征，其中 ○ 和 × 分别表示该列是否使用特定特征。

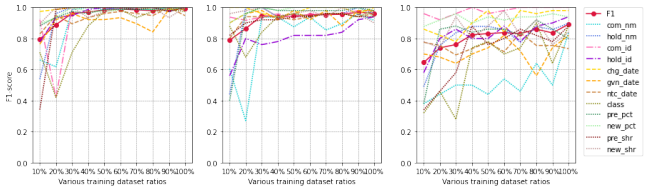
位置编码方法	数字(数)	印刷体($\mathcal{P}$)	手写体($\mathcal{H}$)
不使用位置编码	98.83	94.34	85.98
线性投影	99.09	95.50	86.98
XY位置编码	**99.22**	**96.14**	**89.13**

表6：各种位置编码（PE）的性能

使表征更加全面。在 LXMERT 中也可以发现类似的趋势，其中 T 可能比 VisualBERT 产生更积极的效果（提升约 2%），体现在 $\mathcal{P}$ 和 $\mathcal{H}$方面。对于我们的模型，尽管在 $\mathcal{D}$方面没有观察到明显的提升，但在 $\mathcal{P}$ 和 $\mathcal{H}$ 方面可以发现积极效果，尤其是在 $\mathcal{H}$ 方面（从 84.56 % 提升至 86.98%）。总体而言，所提出的附加特征有助于模型更好地理解表单结构，从而提升模型的泛化能力，尤其是在训练和推理阶段输入表单的外观存在差异时。

7.2.2 位置编码验证。为了展示 XY位置编码 的有效性，我们进行了外部实验，将模型的性能与采用线性投影位置编码方法的模型进行比较，同时也与不采用任何位置编码方法的模型进行比较。表6中，与不采用任何位置编码的模型相比，线性投影和 XY位置编码 在全部三个测试集上都有明显提升。这说明位置信息对于理解表单理解中的布局结构至关重要。此外，从表6可以看出，在 D、P 和 H 上，XY位置编码 的性能优于线性投影。值得注意的是，H 上的 F1分数 从 86.98% 提升到了 89.13%。这表明 XY位置编码可能捕获了更多位置信息，从而更好地理解输入表单中的各种布局结构。

7.2.3 细粒度训练集比例。 为了探究训练集规模对表单理解的影响，我们定义了细粒度训练集比例（从 10% 到 100%）来训练模型，并在图9中展示模型在 D、P 和 H 上的评估性能。随着训练集规模增加，三个测试集上的总体性能和细分性能均呈现波动式提升。在



(a) 数字（$\mathcal{D}$） **(b) 印刷体（$\mathcal{P}$）** **(c) 手写体（$\mathcal{H}$）**

图9：我们的模型在三个测试集上使用细粒度训练集比例时的性能

模型	Acc	细分项（准确率）											
		nm（名称）		id		date			类别	pct（百分比）		shr（股份）	
		com	hold com	持有变更		ntc		gvn		预新	预新		
LXMERT	43.78	86.30	80.14	69.18	52.74	72.60	65.07	54.79	4.79		6.85	24.66	5.48	2.74
VisualBERT	57.48	88.36	78.08	63.01	59.59	39.04	70.55	37.67	47.95		50.00	28.77	53.42	73.29
M4C	69.35	83.56	80.82	63.70	73.29		65.75	57.53	82.88	61.64	65.75	58.90	59.59	78.77
Ours	72.72	91.78	81.51	71.92	71.23		80.82	80.82	84.93	71.92	70.55	80.14	50.00	57.53

表 7：使用 PDFMiner 的关键信息提取任务总体性能及分项结果

达到 50% T 之前，D 和 P 上的性能有明显提升；达到这一关键比例后，则趋于稳定。然而，对于手写体测试集 H，整个训练集规模区间内的总体性能都有显著提升。这可能表明，H 训练数据集的多样性有助于训练出更通用的模型，从而更有效地提取特定关键信息。此外，与提供通用键值标注的数据集不同，我们可以展示特定键值对对训练集规模的敏感性，以分析训练数据有限时模型的鲁棒性。例如，*com_nm*、*com_id*和*hold_nm*对训练集规模更敏感，而与日期相关的键（如通知日期）则不太敏感。_

7.3 使用 PDF 解析器的任务B

现实世界中的用户，尤其是不熟悉深度学习的用户，由于缺乏真实标注和相关背景知识，很难通过训练良好的版面分析模型获取输入感兴趣区域。因此，使用特定 PDF 解析工具提取的文本行，是获取所采用的训练良好模型输入的另一种方式，例如图 10 中的绿色矩形。我们使用 PDFMiner（一种广泛使用的 PDF 解析器 [14]）提取感兴趣区域，以替代数字文档集（D）中的真实感兴趣区域，并将其输入三个经过训练的基线模型和我们的模型。我们使用 IoU =0.5 作为阈值来计算准确率，如表格 7 所示。

与测试阶段输入真实感兴趣区域的趋势不同，未经过预训练的模型 M4C 能取得显著更高的准确率（69.35%），高于预训练模型 LXMERT（35.24%）和 VisualBERT（53.34%）。一种可能的原因是，PDFMiner 检测到的过多噪声感兴趣区域被输入大规模预训练模型，降低了这些大型模型对感兴趣区域的特征表示能力。我们的模型在所有受测模型中取得了最高的总体性能。这表明，所提出的 XY-pos 增强双层模型能够提升通用性，从而更好地理解输入内容。在细分结果方面，*com_nm* 和 *hold_nm* 的表现始终更好。我们的模型提取的大多数

[14]https://pypi.org/project/pdfminer/

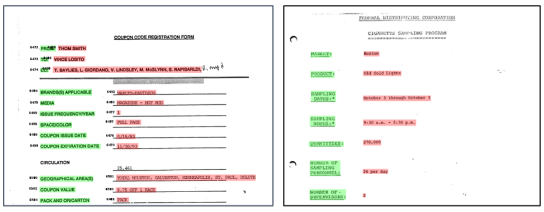


图 10：从 FUNSD 子集中选取的两个文档样本及其答案

关键信息的准确率都高于其他基线模型，例如*com_id* 和 *chg_date*。值得注意的是，LXMERT 对输入感兴趣区域边界框精确率的敏感度很高。然而，由于 PDFMiner 提取的边界框不准确，尤其是在表格区域，因此位于表格中的键的表现远低于其他模型，例如 类别、*pre_pct*、*pre_shr*等。

8 案例研究：迁移学习 NG

为了验证我们所提出的数据和模型的可行性，我们开展了一项迁移学习案例研究。需要注意的是，迁移学习旨在存储解决我们数据集问题时获得的知识，并将预训练模型应用于不同但相关的问题。在本案例研究中，我们将由 Form-NLU 数据集训练得到的最佳模型（我们的模型）和第二佳模型（LXMERT）应用于公开可用的基准数据集 FUNSD。我们使用 FUNSD [10] 数据集中选定的键（问题）-值（答案）对，如图 10 15 所示的样本。根据测试结果，即使 FUNSD 的性质与我们的 Form-NLU 数据集完全不同，在我们的数据集上训练的模型仍然能够取得良好的性能。结果显示，LXMERT（任务B中第二佳的模型）在 150 个 FUNSD 键（问题）-值（答案）对中正确预测了 80 个样本（53.33%），而我们的模型实现了**94（62.67%）**个正确预测。这表明，我们的 Form-NLU 数据集能够学习通用的表单布局，并可应用于其他基准数据集；此外，我们提出的模型能够高效地提取用于表单理解的特征表示。

9 结论

我们提出了 Form-NLU，这是一个用于表单结构理解和关键信息提取的新数据集。该数据集涵盖了一个重要视角，使人们能够理解表单设计者的特定意图，并将用户填写的值与表单对应起来。数据集包含三种表单类型：数字表单、印刷体表单和手写体表单，涵盖多样的表单外观与布局，并能处理其中的噪声。此外，我们提出了一种用于表单结构理解和键值信息提取的强大新基线模型，该模型利用稳健的位置关系和逻辑关系。我们的模型在键值信息提取任务中的表现优于所有最先进的模型。我们希望，所提出的数据集和模型能够为表单结构与信息分析带来重要启发，因此，我们采用了现成的 PDF 布局提取工具，并通过迁移学习验证了其可行性。

[15]请参阅 https://github.com/adlnlp/form_nlu/blob/main/README.md#case-study-setup，查看设置详情。

参考文献

[1] Millicent Chang、Raymond da Silva Rosa 和 Wilson Ng。2016年。收购要约前夕主要股东交易的信息含量。载于亚洲金融协会（AsianFA）2016年会议。[2] Zewen Chi、Heyan Huang、Heng-Da Xu、Houjin Yu、Wanxuan Yin 和 Xian-Ling Mao。2019年。复杂表格结构识别。arXiv 预印本 arXiv:1908.04729（2019年）。[3] Jacob Cohen。1960年。名义量表的一致性系数。《教育与心理测量》20卷第1期（1960年），37–46页。[4] Yihao Ding、Zhe Huang、Runlin Wang、YanHang Zhang、Xianru Chen、Yuzhong Ma、Hyunsuk Chung 和 Soyeon Caren Han。2022年。V-Doc：使用文档回答视觉问题。载于 IEEE/CVF计算机视觉与模式识别会议论文集。21492–21498页。[5] Zhangxuan Gu、Changhua Meng、Ke Wang、Jun Lan、Weiqiang Wang、Ming Gu 和 Liqing Zhang。2022年。Xylayoutlm：面向视觉丰富文档理解的布局感知多模态网络。载于 IEEE/CVF计算机视觉与模式识别会议论文集。4583–4592页。[6] Kaiming He、Georgia Gkioxari、Piotr Dollár 和 Ross Girshick。2017年。Mask R-CNN。载于 IEEE国际计算机视觉会议论文集。2961–2969页。[7] Ronghang Hu、Amanpreet Singh、Trevor Darrell 和 Marcus Rohrbach。2020年。利用指针增强型多模态Transformer进行迭代式答案预测，以完成 TextVQA。载于 IEEE/CVF计算机视觉与模式识别会议论文集。9992–10002页。[8] Yupan Huang、Tengchao Lv、Lei Cui、Yutong Lu 和 Furu Wei。2022年。LayoutLMv3：通过统一文本与图像掩码预训练文档人工智能模型。arXiv 预印本 arXiv:2204.08387（2022年）。[9] Zheng Huang、Kai Chen、Jianhua He、Xiang Bai、Dimosthenis Karatzas、Shijian Lu 和 CV Jawahar。2019年。ICDAR 2019鲁棒阅读挑战赛：扫描收据的 OCR 与信息提取。载于国际文档分析与识别会议。[10] Guillaume Jaume、Hazim Kemal Ekenel 和 Jean-Philippe Thiran。2019年。FUNSD：用于噪声扫描文档表单理解的数据集。载于2019年国际文档分析与识别会议研讨会（ICDARW），第2卷。IEEE，1–6页。[11] Wonjae Kim、Bokyung Son 和 Ildoo Kim。2021年。ViLT：无需卷积或区域监督的视觉与语言Transformer。载于国际机器学习会议。PMLR，5583–5594页。[12] Liunian Harold Li、Mark Yatskar、Da Yin、Cho-Jui Hsieh 和 Kai-Wei Chang。2019年。VisualBERT：一种简单且高效的视觉与语言基线模型。arXiv 预印本 arXiv:1908.03557（2019年）。[13] Minghao Li、Yiheng Xu、Lei Cui、Shaohan Huang、Furu Wei、Zhoujun Li 和 Ming Zhou。2020年。DocBank：用于文档布局分析的基准数据集。载于第28届国际计算语言学会议论文集。949–960页。[14] Siwen Luo、Yihao Ding、Siqu Long、Josiah Poon 和 Soyeon Caren Han。2022年。Doc-GCN：用于文档布局分析的异构图卷积网络。载于第29届国际计算语言学会议论文集。2906–2916页。[15] Minesh Mathew、Dimosthenis Karatzas 和 CV Jawahar。2021年。DocVQA：用于文档图像视觉问答的数据集。载于 IEEE/CVF 冬季计算机视觉应用会议论文集。2200–2209页。[16] Seunghyun Park、Seung Shin、Bado Lee、Junyeop Lee、Jaeheung Surh、Minjoon Seo 和 Hwalsuk Lee。2019年。CORD：用于 OCR 后解析的综合收据数据集。载于 NeurIPS 2019 文档智能研讨会。[17] Johannes Rausch、Octavio Martinez、Fabian Bissig、Ce Zhang 和 Stefan Feuer-riegel。2021年。DocParser：从渲染图像中解析层次化文档结构。载于 AAAI 人工智能大会论文集，第35卷。4328–4338页。[18] Shaoqing Ren、Kaiming He、Ross Girshick 和 Jian Sun。2015年。Faster R-CNN：借助区域提议网络实现实时目标检测。载于《神经信息处理系统进展》第28卷（2015年）。[19] Tomasz Stanisławek、Filip Graliński、Anna Wróblewska、Dawid Lipiński、Agnieszka Kaliska、Paulina Rosalska、Bartosz Topolski 和 Przemysław Biecek。2021年。Kleister：涉及长文档和复杂布局的关键信息提取数据集。载于国际文档分析与识别会议。Springer，564–579页。[20] Hao Tan 和 Mohit Bansal。2019年。LXMERT：学习来自 Transformer 的跨模态编码器表示。arXiv 预印本 arXiv:1908.07490（2019年）。[21] Ryota Tanaka、Kyosuke Nishida 和 Sen Yoshida。2021年。VisualMRC：针对文档图像的机器阅读理解。载于 AAAI 人工智能大会论文集，第35卷。13878–13888页。[22] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Łukasz Kaiser 和 Illia Polosukhin。2017年。Attention Is All You Need。载于《神经信息处理系统进展》第30卷（2017年）。[23] Oriol Vinyals、Meire Fortunato 和 Navdeep Jaitly。2015年。指针网络。载于《神经信息处理系统进展》第28卷（2015年）。[24] Jiapeng Wang、Chongyu Liu、Lianwen Jin、Guozhi Tang、Jiaxin Zhang、Shuaitao Zhang、Qianying Wang、Yaqiang Wu 和 Mingxiang Cai。2021年。迈向稳健现实世界中的视觉信息抽取：新数据集与新颖解决方案。载于*AAAI* 人工智能大会论文集，第35卷。2738–2745。[25] YihengXu、Minghao Li、Lei Cui、Shaohan Huang、Furu Wei 和 Ming Zhou。2020年。LayoutLM：面向文档图像理解的文本与布局预训练。载于第*26*届 *ACM SIGKDD* 知识发现与数据挖掘国际会议。1192–1200。[26] Yiheng Xu、Tengchao Lv、Lei Cui、Guoxin Wang、Yijuan Lu、Dinei Florencio、Cha Zhang 和 Furu Wei。2021年。LayoutXLM：面向多语言富视觉文档理解的多模态预训练。*arXiv* 预印本 *arXiv:2104.08836*（2021年）。[27] Yiheng Xu、Tengchao Lv、Lei Cui、Guoxin Wang、Yijuan Lu、Dinei Florencio、Cha Zhang 和 Furu Wei。2022年。XFUND：多语言富视觉表单理解基准数据集。载于计算语言学协会 *ACL 2022* 会议论文集。3214–3224。[28] Yang Xu、Yiheng Xu、Tengchao Lv、Lei Cui、Furu Wei、Guoxin Wang、Yijuan Lu、Dinei Florencio、Cha Zhang、Wanxiang Che 等。2021年。LayoutLMv2：面向富视觉文档理解的多模态预训练。载于第*59*届计算语言学协会年会和第*11*届自然语言处理国际联合会议（第*1*卷：长篇论文）。2579–2591。[29] Yue Zhang、Bo Zhang、Rui Wang、Junjie Cao、Chen Li 和 Zuyi Bao。2021年。将实体关系抽取视为富视觉文档中的依存句法分析。*arXiv* 预印本 *arXiv:2110.09915*（2021年）。[30] Xinyi Zheng、Douglas Burdick、Lucian Popa、Xu Zhong 和 Nancy Xin Ru Wang。2021年。全局表格提取器（GTE）：利用视觉上下文联合进行表格识别与单元格结构识别的框架。载于*IEEE/CVF* 冬季计算机视觉应用会议论文集。697–706。[31] Xu Zhong、Elaheh ShafieiBavani 和 Antonio Jimeno Yepes。2020年。基于图像的表格识别：数据、模型与评估。载于欧洲计算机视觉会议。Springer，564–580。[32] Xu Zhong、Jianbin Tang 和 Antonio Jimeno Yepes。2019年。PubLayNet：迄今最大的文档布局分析数据集。载于*2019*年国际文档分析与识别会议（*ICDAR*）。IEEE，1015–1022。