Semantic Recognition on Table Images from Visually Rich Documents

by

Bin Xiao

Thesis submitted to the University of Ottawa
in partial fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Declaration of Authorship

I hereby certify that this thesis is entirely my own original work except where otherwise indicated. I am aware of the University's regulations concerning plagiarism, including those concerning consequent disciplinary actions. Any use of the works of any other author, in any form, is properly acknowledged at their point of use.

Abstract

Visually rich documents have been widely used in many scenarios because of their user-friendliness for human readers. However, with the surging number of these documents, it has been far beyond the capacity of humans to manage, extract critical information and mine useful knowledge from these documents efficiently, making it necessary to develop tools to manage and interpret these documents automatically. Typically, a document can contain different types of components, such as Regular Text areas, Tables, and Figures, and these components usually require different processing methods. Since tables are usually used to summarize vital information, and their two dimensions and complex structures make the semantic recognition challenging, this thesis focuses on the semantic recognition task on tables. With the development of Large Language Models (LLMs), applying LLMs to semantic recognition tasks has become a popular choice, as many studies have demonstrated the remarkable capacities of LLMs for semantic recognition tasks. Considering that even though multi-model LLMs can process images directly, their performance on semantic recognition tasks with images containing dense text is still far behind text-only LLMs, this thesis proposes to apply text-only LLMs on semantic recognition task on the table images from visually rich documents. Since the tables from visually rich documents are usually images, this thesis introduces a complete solution to fill the modality gap between table images and text-only LLMs, including Table Detection (TD) and Table Structures Recognition (TSR) models, and further use Table Question Answering (Table-QA) problem as an example of semantic recognition tasks. Comprehensive experiments are conducted on various datasets for models in the TD, TSR and Table-QA, and the experimental results demonstrate the superiority of the proposed solution.

Dedication

First and foremost, I would like to express my deepest gratitude to my advisor, Professor Burak Kantarci, for his invaluable guidance, support, and encouragement throughout the completion of this thesis. His expertise, insight, and patience have been instrumental in shaping this work, and I am immensely grateful for the time and effort he dedicated to helping me grow academically and personally.

I would also like to extend my heartfelt thanks to my family for their unwavering love, understanding, and encouragement. To my parents, thank you for always believing in me and providing me with the foundation to pursue my dreams. Your endless support has been a constant source of strength, and I could not have come this far without you.

Lastly, I would like to thank my friends and colleagues who have supported me during this journey. Your encouragement and companionship made this challenging process much more manageable and fulfilling.

To everyone who has played a part in this achievement, I am deeply thankful.

Table of Contents

List of Tables

List of Figures

List of Symbols

$A \cap B$ the intersection of set A and B

$A \cup B$ the union of set A and B

C the smallest enclosing convex

Δp_n the nth learnable offset

$\mathcal{L}$ loss function

$\mathcal{N}(\mu, \sigma^2)$ Gaussian Distribution with μ as mean value and σ^2 as variance

$\{b_i, c_i\}$ the ith bounding box and its corresponding object class

$\{c_x, c_y, w, h\}$ a bounding box represented by its center in x, y axis and its width and height

p_n the nth grid point in grid R

$w(p_n)$ the weight at the nth grid point

List of Abbreviations

CoT	Chain of Thought
TD	Table Detection
$Table - QA$	Table Question Answering
$TEDS$	Tree-Edit-Distance-Based Similarity
TSR	Table Structure Recognition
$HTML$	HyperText Markup Language
ICL	In Context Learning
ICS	Information Coverage Score
ICT	Information and communication technologies
IoU	Intersection over Union
LLM	Large Language Model
mAP	Mean Average Precision
NMS	Non Maximum Suppression
PDF	Portable Document Format
RPN	Region Proposal Network

Chapter 1

Introduction

Portable Document Format (PDF) and scanned documents are the dominant data formats in business scenarios and on the internet. However, these documents cannot provide enough metadata to fully describe their contents and structures. With the surging number of these documents, it has been far beyond the capacity to manage and mine knowledge from these documents for human readers, making it necessary to develop tools to manage, interpret, and understand them. Generally, a visually rich document can combine several components, such as Text Area, Figure, Formula, and Table. Among these components, Tables are usually used to summarize critical information but are challenging to interpret because a table's structure can be very complex and cannot be processed as a flat sequence. Therefore, this thesis focuses on the semantic recognition problem of table images from visually rich documents and proposes a complete solution to extract, process and interpret the tables from the visually rich document images. More specifically, as shown in Figure 1.1, the proposed solution formulates the problem as a Table Detection (TD) task, a Table Structure Recognition (TSR) task and a Table Question Answering (Table-QA) task. TD aims to detect and extract tables from document images, and its results are further processed by the TSR model and the OCR model to transform the table images into structured or semi-structured text formats, such as HTML sequences. At last, the converted text sequences are the inputs of the Table-QA model as the semantic recognition task. The Table-QA model in this pipeline only needs to take text-only inputs, making it easier to utilize large language models (LLMs). This thesis employs detection models for the TD and TSR tasks, which can be trained end-to-end, and uses a prompting method for the Table-QA task without training.

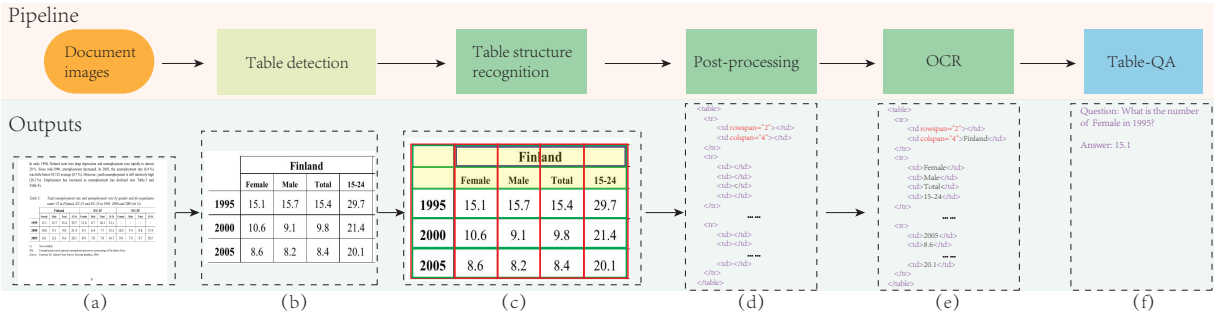


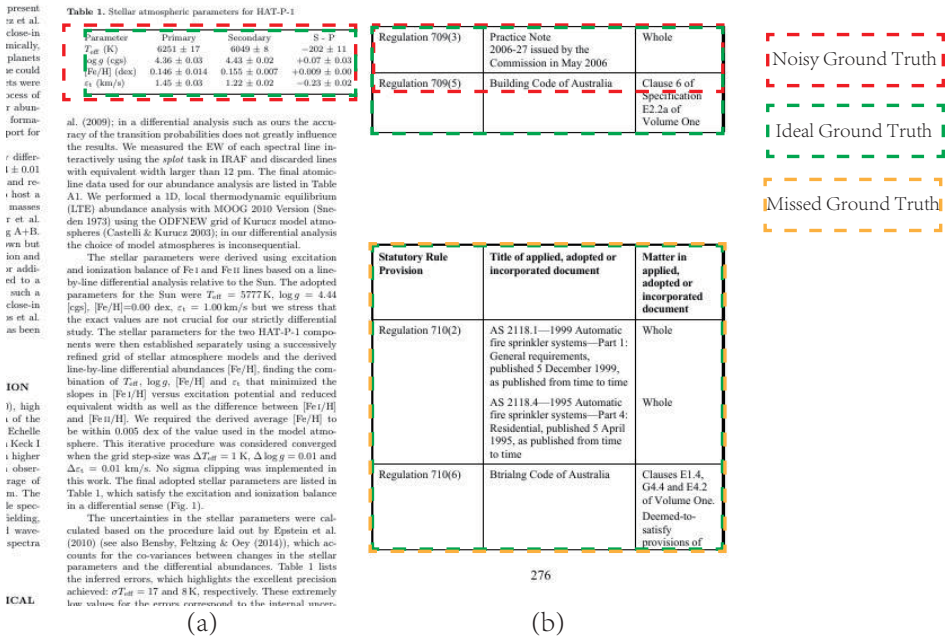
Figure 1.1: Workflow of the solution proposed in this thesis. This solution contains TD and TSR to extract and transform table images from visually rich documents into a structured or semi-structured text format, such as HTML sequence, and then use text-only LLMs for the Table-QA task.

1.1 Table Detection

TD is the first task for the proposed pipeline to locate and extract tables from the visually rich documents, as shown in Figure 1.1. Many studies have proposed datasets and solutions for the TD problem. However, existing datasets either suffer from limited samples or are too noisy. For example, ICDAR2013 [1], ICDAR2017 [2], and ICDAR2019 [3] are manually labeled, which means that the annotations are more reliable and consistent, but the number of training sample in these datasets are usually limited. TableBank [4] and PubLayNet [5] are annotated by parsing metadata of electronic documents, making these annotations noisy and inconsistent, even though these datasets are much larger. Besides, the annotation definitions across these datasets are often different, which means we cannot simply merge these datasets together to form larger datasets. Figure 1.2 shows two samples from the TableBank test set. One typical issue of these meta-data generated datasets is that the bounding box can be larger than an ideal bounding box, as shown in Figure 1.2 (a), which can make the evaluation unreliable when the Intersection over Union (IoU) threshold is high. Another issue is that some tables are missed or the bounding box is not large enough to cover the whole table, as shown in Figure 1.2 (b). The quality of a table detection set is critical for the TD problem because a successful TD application should avoid losing information presented in the tables. The noisy labels in the test set can influence the model evaluation, especially for widely used evaluation metrics threshold by IoU scores. It is worth mentioning that even though manually annotated datasets have a higher quality of annotations, there are still many noisy samples in both their training and testing sets. Therefore, this thesis revisits several well-annotated datasets, including ICDAR2013 [1],

ICDAR2017 [2], ICDAR2019 [3], Marmot [6] and TNCR [7], aligns the labeling definition of these datasets, cleans the noisy samples and merge them together to form a larger dataset, termed with Open-Tables. The new Open-Tables dataset can minimize the side effects of noisy samples on the model evaluation and provide more reliable results. Besides the issues of noisy labels, the data sources of open datasets are limited, primarily from academic publications or public governmental documents, making the intra-class variance and the inter-class variance of these documents small because these documents have to be written following a series of writing principles. In other words, detection models can easily achieve a promising performance on these datasets, which also cannot reflect the complexity of real enterprise applications. Therefore, this thesis proposes a new TD dataset using datasheets from the Information and Communications Technology (ICT) domain. It is a more challenging dataset because of the domain-specific samples, layout and appearance variances. Figure 3.3 shows some samples from our proposed dataset, which can hardly be found in the public datasets. For example, Figure 3.3 (a) is a special table containing several sub-tables, and Figure 3.3 (7) is a table containing figures as the content of some table cells. More details regarding the proposed datasets are included in Chapter 3.

On the other hand, Table Detection (TD) is typically formulated as an object detection problem defining tables as the detection targets. Current state-of-the-art methods [8,9] for the TD problem usually employ two-stage object detectors, which require dense candidates and apply data augmentation and multiple-stage transfer learning techniques. However, tables in visually rich documents are usually well formatted and large so that human readers can easily interpret them. Besides, the number of tables in a single document image is typically small, which means their distribution in a single document is sparse. Based on these observations, this thesis uses SparseR-CNN [10] as the base model, which is a competitive detector using sparse learnable regional proposals. It is worth mentioning that many state-of-the-art studies for the TD problem often employ two-stage detectors using dense candidates and multiple-stage transfer learning techniques, which are usually more complex than the proposed method. This thesis also proposes using image-size regional proposals to cover all the information on target tables in the proposal boxes and using the noise augmentation method to enrich the diversity of proposal boxes. Since information extraction tasks often follow TD tasks, it is vital to avoid information loss. Therefore, a larger prediction box is preferable to a smaller box that can lose information even when the latter box has a larger IoU score with the ground truth. Figure 1.3 uses a table as an example to further illustrate this observation. The green box in Figure 1.3 has an IoU score of 0.77 with the red ground truth box, while the blue box has an IoU score of 0.82. Even though the blue prediction has a larger IoU score, the green prediction is preferable for TD tasks. These observations show that the IoU score cannot directly reflect the information



(a) (b)

Figure 1.2: Two noisy samples from the TableBank test dataset. Red bounding boxes are the ground truth boxes provided by the dataset. In Figure (a), the bounding box is larger than the ideal bounding box, which can make the evaluation unreliable when the IoU threshold is high. Figure (b) shows a sample in which the bounding box is not large enough, only covering part of the table. Besides, the table at the bottom of Figure (b) is missing. It is worth mentioning the images' low resolution is caused by the images provided by the dataset.

loss of a prediction box. Therefore, this thesis proposes to decouple the IoU score into two terms: a ground truth coverage term and a prediction coverage term, in which the former term can be used to measure the information loss for the prediction boxes. It is worth mentioning that the proposed decoupled IoU score termed the Information Coverage Score (ICS), can replace the IoU score in the IoU-based loss functions and evaluation metrics. Besides, label assignment in object detection models is to define the classification and regression targets for anchors [11]. Many studies [11, 12] have shown that label assignment plays a vital role in the success of a detector, and the one-to-one scheme used in SparseR-CNN [10] is not optimal. SparseR-CNN employs a cascade architecture that uses the outputs of ith Dynamic Head as the inputs of the $i+1$th Dynamic Head to refine the predictions, as shown in Figure 4.1, which means that the proposal quality of each Dynamic Head varies. Therefore, inspired by the studies [11–15], this thesis leverages a SimOTA [15]

based many-to-one label assignment approach to further improves the SimOTA by adopting a dynamic scheduling scheme to adjust the number of positive assignments dynamically and integrating the proposed ICS loss to the cost function. The details of the proposed SparseR-CNN-based TD model and the decoupled IoU are discussed in Chapter 4.

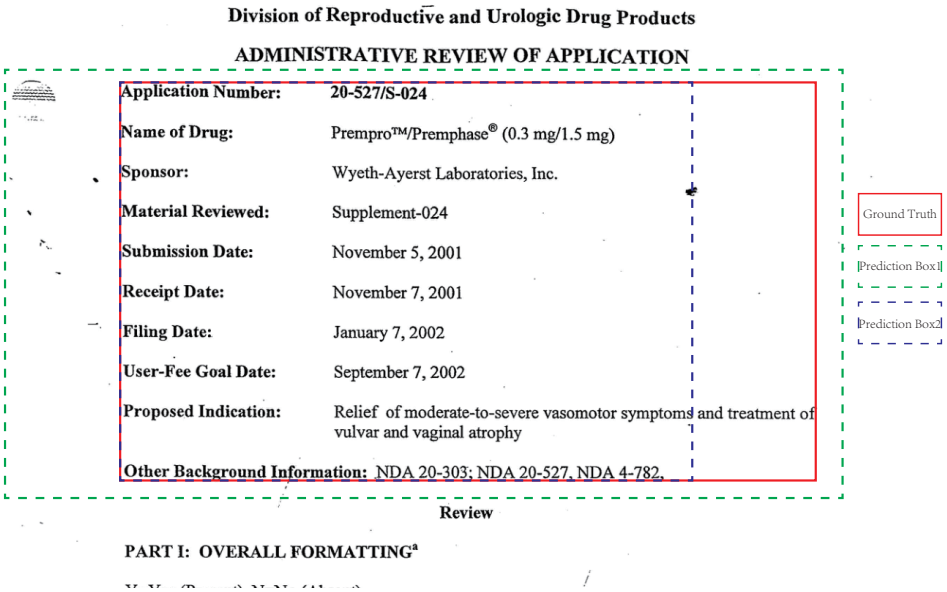


Figure 1.3: Preferable prediction for TD tasks. The IoU scores of green and blue predictions are 0.77 and 0.82, respectively. Even though green prediction's IoU is smaller, it is preferable for TD tasks.

1.2 Table Structure Recognition

After the TD task in the proposed solution, as shown in Figure 1.1, TSR aims at transforming table images into a structured format, such as an HTML sequence. TSR studies can roughly be categorized into three groups based on their problem formulations: image-to-sequence, graph-based, and detection-based models. Image-to-sequence models usually follow the encoder-decoder architecture and directly generate structured outputs, such as HTML sequences. Some image-to-sequence models [16, 17] also integrate the OCR task into the model to make the model end-to-end without using extra OCR tools [18, 19] to extract text contents from the images. However, since these models use auto-regressive

decoders, they often suffer from error accumulation [20], and their OCR capacity usually cannot generalize well because of the limitation of training data. On the other hand, graph-based models usually use segmentation or detection methods to extract table cells, treat extract table cells as nodes of a graph, and further build the relation among the graph nodes. This graph-based definition makes it easier to deal with the scenarios in which table images are collected from the wild, such as rotated, distorted tables. However, graph-based models introduce extra complexity because they need to build extra graph models compared with detection-based models. By contrast, detection-based models are more straightforward in detecting the table components directly and post-processing the detection results with a deterministic rule-based method for reconstructing the table structure. However, detection-based methods can fail to deal with rotated and distorted samples. Besides, detection-based models usually cannot perform as well as other types of solutions regarding cell-level TSR metrics, such as TEDS [21]. Therefore, these different types of approaches have their benefits and must be selected based on the application scenarios. This thesis focuses on applying detection-based TSR models to process the table images from well-formatted documents.

There have been many studies [22–24] using detection models together with a post-processing method to solve the TSR task. However, existing studies either over-simplify the problem or define a multi-label detection task, which is challenging for two-stage object detectors. For example, some studies [22, 23] do not define the Column Header as a detection target, making it impossible to provide information regarding the header cells. By contrast, PubTables1M [24] defines six types of components, including Table, Column, Row, Spanning cell, Column Header, and Projected Row Header, which can provide as much structure information as other types of TSR models. However, PubTables1M [24] does not consider that some Column Headers and Projected Row Headers can share identical bounding boxes with corresponding Rows, making this definition a multi-label detection task. Besides these issues of problem formulation, detection models used in these studies are trained to optimize their detection performance. However, since the complex structures of tables are processed by a post-processing step inferring defined table components, such as Columns and Rows, a model with good detection performance cannot necessarily lead to good performance in TSR metrics, such as TEDS. Moreover, some critical characteristics of detection models are not considered in existing studies' model design and problem formulation. For example, typical two-stage detection models, such as Cascade R-CNN [25], are not suitable for multi-label detection tasks, while transformer-based detection models, such as DETR [26] and SparseR-CNN [10], can achieve promising results on multi-label detection tasks. Another example is that for two-stage detection models, regional proposal generation plays a crucial role in the model's performance, and the defined components in

table images have different aspect ratios compared with common objects. At last, many studies apply deformable convolution [27] to improve the models' performance regarding detection evaluation metrics, such as COCO metric [28]. However, simply applying deformable convolution can degrade the model's performance regarding the TEDS, and it is necessary to extract long-range dependencies while improving the local feature extraction. Therefore, this thesis comprehensively revisits existing detection-based solutions and further explores the possible reasons hindering the performance of detection-based models for the TSR task. Based on the findings and analysis, this thesis applies three simple methods to a typical two-stage detection model, Cascade R-CNN, including tuning the aspect ratios and increasing the number of region proposals in regional proposal generation, transforming the multi-label detection task into the single-label task, and introducing a Spatial Attention Module to build long-range dependencies. The experimental results show that the proposed method can achieve state-of-the-art performance with very simple methods, demonstrating that the findings can be a guideline for further improvement of detection-based solutions. More details regarding the proposed detection-based TSR solution are discussed in Chapter 5.

1.3 Table Question Answering

Table Question Answering (Table-QA) is the last task in the proposed solution, which is a typical semantic recognition task. Typically, tables can be easily categorized into two groups: structured and semi-structured tables. Structured tables are usually from relational database systems with explicit schema describing their structures and data types, meaning the programming languages, such as SQL, can naturally process them. By contrast, tables from other sources, such as web pages and visually rich documents, are usually semi-structured without schema requirements, resulting in complex structures and heterogeneous data types, making the QA task on these semi-structured tables more challenging. This study focuses on the Table-QA problem on the semi-structured tables, which is a more challenging setting.

As pointed out by many studies [29–31], Large Language Models (LLMs) suffer from some limitations in their arithmetic calculation and complex reasoning capacities. Therefore, many prompting tuning methods have been proposed to enhance LLMs' capacities. For example, Chain of Thought (CoT) [31] is a popular prompting method providing reasoning rationales to trigger LLMs' reasoning capacities. PAL [29] and PoT [30] propose to offload the reasoning steps to Python code. However, these typical prompting methods cannot perform well in the semi-structured Table-QA problem because of the complex

structures, heterogeneous data types, and sometimes huge tables with numerous columns and rows. More specifically, prompt methods without applying programming languages must first extract the correct information from semi-structured tables before reasoning, which is challenging for LLMs, especially when the table is huge [32]. Figure 1.5 contains a failure example of CoT, which interprets *1936/37* as two years and fails to extract the correct *Year 1953/54*. Similarly, programming-aided solutions, such as PAL and PoT, can also suffer from this information extraction issue if we define the relevant information as Python variables. Applying Pandas Library [33, 34] can alleviate this information extraction issue [35] by providing proper selection criteria, but introducing extra difficulties caused by the heterogeneous data types. Figure 1.5 shows a PoT example using Pandas Library, which fails to run because the values in column *Year* cannot be directly compared with *1936*. Besides, complex structures of semi-structured tables can often lead to wrong results, especially for program-aided solutions. Figure 1.4a shows an example from WikiTableQA [36] dataset, which contains several spanning cells across multiple columns and rows, and inconsistent data types, such as the column *Season*. Even though some methods [24] can transform this table into a standard table by repeating table spanning cells into multiple single table cells so that SQL or Python Pandas library can process it, its structure still can lead to wrong results. For example, when an LLM is prompted to generate Python code to answer the question *"What is the maximum League Apps after 2004?"*. Two *Total*s in the column *Season* will be compared with correct *Season*s, which leads to wrong results. Besides, the generated code needs to compare the values from the column *Season*, which can lead to an error of execution because " $>$ " operation cannot be applied to the string *"2011-12"* and the integer *2004*, as highlighted in Figure 1.4b, which is another failure example caused by the heterogeneous data types. Along with applying Python to enhance the LLMs, some studies [37, 38] apply SQL to the Table-QA problem. However, SQL is a programming language designed for structured tables, meaning that semi-structured tables need to be transformed into structured format first so that tables can be imported into the relational databases to execute SQL queries, which is another challenging task for the semi-structured Table-QA problem discussed in this study. Besides, as pointed out by some studies [37, 39], some questions are not answerable by merely using SQL. Therefore, considering the flexibility of Python and the limitations of applying SQL to the semi-structured Table-QA task, Python is applied in this thesis.

Besides the issues of applying prompting methods to the semi-structured Table-QA problem, current studies only focus on optimizing the prediction accuracy without considering their inference cost. As mentioned, some tables can be huge, which can lead to long prompts and inference time. Even though some solutions [38, 40] propose decomposing huge tables into sub-tables for further reasoning, they need to prompt the full huge table

Club	Season	League		Cup[1]		Continental[2]		Other[3]		Total	
		Apps	Goals	Apps	Goals	Apps	Goals	Apps	Goals	Apps	Goals
Anzhi Makhachkala	1999	40	2			-		-		40	2
	2000	30	3			-		-		30	3
	2001	14	0			-		-		14	0
	Total	**84**	**5**							**84**	**5**
CSKA Moscow	2001	12	1			-		-		12	1
	2002	29	2			2	0	-		31	2
	2003	28	1			2	0	1	0	31	1
	2004	26	1			10	0	1	0	27	1
	2005	30	1	0	0	15	0	-		45	1
	2006	30	1	6	0	7	0	1	0	44	1
	2007	27	0	5	0	7	0	1	0	40	0
	2008	23	0	3	0	-		-		18	0
	2009	10	0	1	0	6	0	1	0	18	0
	2010	11	0	1	0	6	0	0	0	18	0
	2011–12	13	0	1	0	1	0	0	0	15	0
	2012–13	0	0	1	0	0	0	-		1	0
	2013–14	1	0	0	0	0	0	-		1	0
	Total	**240**	**7**	**1**	**0**	**56**	**0**	**5**	**0**	**302**	**7**
Career total		**324**	**12**	**1**	**0**	**56**	**0**	**5**	**0**	**386**	**12**

(a) A sample table with a complex structure from the WikiTableQA dataset.

```python
#solution in Python
import pandas as pd
def solution(table_dict):
    df = pd.DataFrame(table_dict)
    # Filter the rows after 2004
    df_after_2004 = df[df['Season'] > 2004]
    # Find the maximum League Apps
    max_league_apps = df_after_2004['League Apps'].max()
    return max_league_apps
```

(b) A sample of generated Python code to answer the question *What is the maximum League Apps after 2004?*.

Figure 1.4: An example of a semi-structured table and its failed Python code because of the heterogeneous data types and the table's complex structure. The Python code is generated by Mixtral-8*7B.

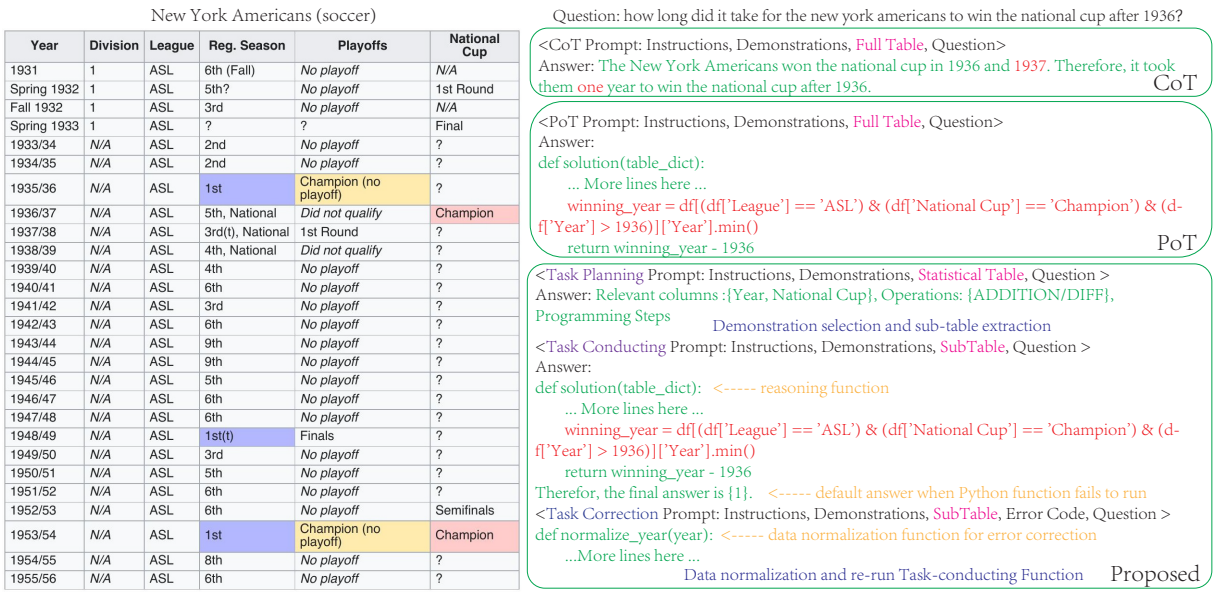


Figure 1.5: Comparison of CoT, PoT and the proposed method. It is worth mentioning that many details are omitted due to space limitations. The details of the prompts are attached in Appendix .4.

into the LLM first. Some studies [32] truncate the huge tables, which can drastically reduce the inference cost but introduce the risk of losing critical information in the tables and sometimes make it impossible to give the correct answer. Besides, ensemble methods, such as self-consistency decoding and majority voting, are often employed to improve the performance further [38, 40, 41], but also drastically increasing the inference cost simultaneously. Moreover, current studies [30,37,38,40,42–44] are usually built upon Open-AI's commercial LLMs, which is not practical for many scenarios with data security and hardware considerations. Even though some open-source LLMs can show promising performances on various public benchmarks [45, 46], these open-source LLMs can show different behaviours from state-of-the-art commercial LLMs, because of their number of parameters, training and instruction tuning datasets, and many other factors. For example, study [32] demonstrates that CoT [31] can improve the prediction accuracy on the Table-QA task with GPT3 [47]. However, our experiments show that the benefits of CoT [31] are incremental with various open-source LLMs with different scales of parameters and released by multiple companies.

Lastly, most of these studies are based on In Context Learning (ICL), using demonstrations to guide the LLM in conducting target tasks, while crafting and selecting proper demonstrations is still an open issue. Many studies [48, 49] pointed out that the content, number and order of demonstration can all influence the results. Therefore, a series of atomic operations is first defined to measure the complexity of a query question in the given table and describe the steps with the defined atomic operations, which can be a metric for the demonstration selection when crafting prompts. To mitigate the issues limiting the performance of LLMs, including the complex table structure, heterogeneous data types, huge tables, and the inherent limitations of LLMs in reasoning capacities, a three-stage prompting solution is proposed containing task-planning, task-conducting and task-correction stages, as shown in Figure 1.5. The task-planning stage prompts the LLM to analyze the statistical information of the given table and provide programming steps, data requirements, and relevant columns to solve the query question and generate a plan. Then, the task-conducting stage first generates a default answer as the final answer when the generated Python code fails to execute and then generates the Python code based on the plan from the first stage. When a task-conducting stage fails to execute, the heterogeneous data types are usually the reason. Therefore, we also include a task-correction stage, which can generate normalization functions to normalize the data and correct the error in the task-conducting stage. It is worth mentioning that the proposed method can avoid huge tables as a part of the prompt, which can drastically reduce the number of prompting tokens when the tables are huge. More details regarding the proposed Table-QA solution are discussed in Chapter 6.

It is worth mentioning that, as shown in Figure 1.1, an OCR tool is needed to extract

text content from the images so that the table images can be converted into structured or semi-structured text formats. Since there have been many OCR tools, such as EasyOCR [19] and MMOCR [18], achieving very promising performance, this thesis simply employs open-source OCR tools for text extraction.

1.4 Contributions

To sum up, the contributions of this thesis are four-fold:

1. This thesis revisits existing open-source datasets, addresses the issues of inconsistent annotations, noisy annotations and limited data sources, and proposes two highly high-quality datasets by merging cleaned open-source datasets and annotating collected data from the Information and Communications Technology (ICT) domain, termed with Open-Tables and ICT-TD datasets.

2. This thesis proposes a Sparse R-CNN based TD solution, incorporating a series of methods to improve its performance, including adapting a dynamic scheduling scheme to the label assignment, proposing a decoupled IoU loss and applying a Gaussian Noise Augmented Image Size region proposal method.

3. This thesis revisits existing detection-based TSR solutions and further analyzes the limitations of existing solutions, including the over-simplified problem formulation, the mismatch of detection and TSR metrics, the mismatch of multi-label detection formulation and the detection models. After the analysis of these issues, this thesis proposes three simple methods to improve a Cascade R-CNN-based method, including tuning the aspect ratios and the number of region proposals, extending the multi-label detection to regular single-label detection method by generating pseudo-class, and a Spatial Attention Module to build long dependencies.

4. This thesis formulates a Table-QA task, taking the results of the TSR task and Questions as inputs, and further explores programming language-aided prompting methods to utilize the remarkable abilities of LLMs. More specifically, a series of atomic operations are defined to describe and measure the similarities of QA tasks, which can be a guide to craft demonstrations and a distance metric for demonstration selection. Besides, a three-stage prompting solution is proposed, including task-planning, task-conducting and task-correction stages, to mitigate the issues caused by the semi-structured tables and the limitations of LLMs.

The rest of this thesis is organized as follows: Chapter 2 discusses related studies, including object detection models, table detection datasets, table detection models, table structure recognition, pre-trained language models and large language models. Chapter 3 revisits open-source TD datasets and introduces proposed Open-Tables and ICT-TD datasets. Chapter 4 introduces the proposed SparseR-CNN based model for the TD task. Chapter 5 revisits existing detection-based TSR solutions and presents the proposed Cascade-RCNN based TSR model. Chapter 6 discusses the proposed methods for the Table-QA task. At last, conclusions and future directions are discussed in Chapter 7.

Chapter 2

Related Work

Since both TD and TSR problems are formulated as Object Detection problems in this thesis, this chapter first discusses the state-of-the-art Object Detection models, then discusses tasks and related datasets used in this thesis, including TD, TSR, Pre-trained Language Models, Large Language Models, LLM-based Table-QA Methods, Table-QA datasets and Prompting Methods. It is worth mentioning that this section focuses on machine learning and deep learning approaches because these methods have become the dominant solutions for these tasks.

2.1 Object Detection Models

Object detection problem has been widely discussed in recent years. Object detection models are often categorized into one-stage and two-stage models based on their number of regression steps. Popular one-stage models includes YOLO [50] and its variants [15,51–53], SSD [54], FCOS [55], and many others. These one-stage models do not contain the step of generating region proposals. For example, YOLO series models divide the images into grids first, then classify the class of grid cells and directly predict the bounding boxes and their confidences. The simple design of one-stage detectors leads to faster training and inference time compared with two-stage detectors. In contrast, two-stage models usually first use a region proposal network to generate a series of regional proposals, then further regress and classify the regional proposals by well-designed models. Typical two-stage models includes Faster-RCNN [56], MaskR-CNN [57], CascadeR-CNN [58] and many others.

On the other hand, some studies [10, 59] categorize the popular detectors from the

perspective of Non-maximum Suppression (NMS), which is widely used to reduce the redundant predictions from the detectors. From this perspective, popular detectors can be categorized into end-to-end and none end-to-end models based on whether NMS is needed. DETR [26] is a typical end-to-end detector introducing transformer architecture [60], set prediction loss, and one-to-one label assignment to the object detection problem. Following the design of DETR, many variants of DETR have been proposed to improve the performance further and accelerate the convergence, such as Deformable-DETR [61] and DAB-DETR [62]. Sparse R-CNN [10] refactors the DETR model and proposes to use sparse learnable regional proposals to replace dense regional proposals and utilize a dynamic instance interactive head to regress and classify the proposals in an iterative manner. Study [63] analyses the success of end-to-end detectors and argues that the one-to-one label assignment method in end-to-end detectors contributes to the success of the end-to-end models but is not sufficient to fully remove the NMS from the pipeline. This study further points out that the classification cost in the matching cost when applying one-to-one label assignment plays a key role in the success of these end-to-end models. Study [59] further analyzes combinations of the label assignment methods and queries and argues that sparse queries with one-to-one label assignment can degrade the recall, and dense queries with one-to-one label assignment are hard to optimize. To address these issues, study [59] proposes a dense distinct queries (DDQ) method to select distinct queries from dense queries using a class-agnostic NMS, achieving promising precision and recall. SQR [64] points out that the stages in DETR series detectors have different unbalanced responsibilities and proposes to collect and select intermediate queries for subsequent stages. It is worth mentioning that these end-to-end detectors can easily be extended to none end-to-end solutions by adapting many-to-one label assignments [12] and NMS. This thesis refers to models using transformer architecture, set prediction loss, and their variations as transformer-based detection models, such as DETR, Sparse R-CNN and Deformable-DETR [61], and the proposed method in this thesis is based on SparseR-CNN and uses the image size initialization considering the requirements of TD applications as discussed in Chapter 1.

Besides these object detection models, loss functions used in object detectors' regression and classification tasks have also been discussed widely. For the regression task, it is a natural choice to use l_n-norms and their variants, such as smooth-l1 [65], as the loss function. However, these functions are not aligned with the widely accepted evaluation metric IoU score, meaning that for some cases, minimizing these loss functions cannot lead to better IoU scores [66, 67]. IoU-based loss functions can alleviate this issue and become the most popular choice. IoU loss [68] has the gradient vanishing issue when the prediction and ground truth boxes have no overlaps. GIoU loss [66] extends the IoU loss by adding an extra penalty term to alleviate the gradient vanishing problem when two boxes have no

overlap. More specifically, assuming that A, B denote two arbitrary convex shapes and C is the smallest enclosing convex, then the term used in GIoU loss is defined as $\frac{|C-A\cup B|}{|C|}$, where $-$ means the complementary operation. A limitation of GIoU loss is that it can be degraded to IoU loss for enclosing bounding boxes. To address this limitation of GIoU loss, DIoU loss [69] proposes to use the distance between two boxes' centers as the additional term, which can lead to faster convergence and alleviate the gradient vanishing problem. CIoU loss [70] also considers the geometric factors of bounding boxes and proposes an aspect ratio term, a distance term, and the IoU term. Besides these popular IoU-based loss functions, there are other loss functions without IoU, such as SCALoss [67] and KLLoss [71]. SCALoss defines two terms considering side overlap and corner distance. KLLoss requires the output to be a distribution instead of location coordinates.

At last, label assignment methods are another widely discussed aspect of detection models. Typically, label assignment methods can be categorized into Fixed Label Assignment and Dynamic Label Assignment [11]. Fixed Label Assignments are methods defining fixed criterion to determine the positive and negative samples of each ground truth. For example, Region Proposal Network (RPN) in FasterR-CNN [56] defines two IoU scores as the thresholds to the positive and negative proposals. YOLO [50] uses the closest anchor points to the center of ground truths as positive anchor points. In contrast, Dynamic Label Assignment methods often formulate the problem as an optimization problem and solve the problem more dynamically. For example, OTA [11] formulates the label assignment problem as an Optimal Transport (OT) problem, which can be optimized by the Sinkhorn-Knopp algorithm. SimOTA [14, 15] uses the tok-K candidates whose centers are in the ground truth bounding boxes to avoid the time-consuming optimization process.

2.2 Table Detection Datasets

There have been many public datasets that can be used for the TD problem. Some of these datasets are created only for the TD problem, such as ICDAR2013 [1], ICDAR2017 [2], ICDAR2019 [3] and TNCR [7]. The ICDAR2013 dataset is a widely used benchmark in many studies, and it contains 238 images collected from public governmental documents. Some images in the ICDAR2013 dataset do not contain tables, resulting in 150 tables. Since this dataset is relatively small, many studies have achieved 100% F1-score, following the evaluation metric setting of ICDAR2013 competition [1] whose IoU threshold is 0.5. The ICDAR2017 and ICDAR2019 datasets are the other two datasets used in the respective competitions. The ICDAR2017 dataset consists of a training set with 1600 images and a testing set with 817 images. The documents in the ICDAR2017 are collected from CiteSeer,

which is a source of academic publications. ICDAR2019 has two separate collections, including archival and modern document images. This thesis only considers the modern document set of the ICDAR2019 dataset, which contains 600 images for training and 240 for testing. TNCR [7] also collects the Public Governmental Documents and further defines five types of tables, including Full-lined, No-lined, Merged-cells, Partial lined, and Partial lined merged cells. TNCR contains 4634, 987, and 1000 images for training, testing and validation, respectively. IIIT-AR-13K [72] is another dataset defining five types of document objects, including Table, Figure, Natural Image, Logo and Signature. IIIT-AR-13K dataset uses business annual reports as the data source and consists of 9000 training images, 2000 testing images and 2000 validation images.

There are also some datasets generated by parsing the meta-data of Microsoft Word files or latex sources, which are often much larger than manually annotated datasets. TableBank [4] is a typical automatically generated dataset containing two parts: latex-generated samples and Microsoft Word-generated samples. The Word files are collected from the internet, and the latex source codes are from open academic publications. Similarly, PubLayNet [5] dataset is generated by parsing the XML format documents of academic publications. PubLayNet defines five types of document components, including Text, Title, List, Table and Figure. The statistics of these open datasets are summarized in Table 3.2.

As discussed in Chapter 1, the open datasets discussed here have some obvious limitations. Firstly, the annotation definitions across these datasets can be different, especially for the ambiguous samples. Second, the data sources of these datasets are limited, mainly from academic publications and open governmental documents. Even though there are other data sources, such as business annual reports, the samples from these sources are usually simple and similar to those from other open datasets. Third, these open datasets can be noisy, especially for these automatically generated datasets, which can make the model evaluation using these datasets are not reliable. Besides, the data sources of these datasets are very limited, mainly from academic publications and open governmental documents. These limitations make the models trained with datasets hardly have good generalization ability to the different domains. Besides, the noise samples in the training set can hinder the model performance, and the noise samples in the testing set can make the model evaluation unreliable. Therefore, to alleviate these limitations, this thesis proposes the Open-Tables and ICT-TD datasets, which have consistent annotation and less noise. Besides, this thesis also builds the benchmarks for the cross-domain setting, which is challenging but useful for TD applications.

2.3 Table Detection Models

Table Detection is a fundamental step for downstream tasks such as key information extraction and visually rich document understanding. Many studies have recently discussed the TD problem. One of the most popular formulations for the TD problem is defining Tables in visually rich documents as objects and then applying popular object detectors. Following this problem formulation, the object detection approaches discussed in section 2.1 can be easily adapted to the TD problem and widely used as benchmark models in many studies [7]. Considering the special characters and requirements for the TD problem, many studies further optimized the popular object detection methods to improve the model performance. Due to the limited number of training samples for the TD problem, transfer learning methods are widely used. CascadeTabNet is based on Cascade Mask R-CNN [73] with HRNet [74] as the backbone network. In addition, CascadeTabNet also employs an image augmentation method, which can thicken the text regions and reduce the areas of blank space, applying a two-stage transfer learning approach and various augmentation methods to train the model in an iterative manner. Similarly, TableDet [8] is based on Cascade R-CNN [58], proposes a Table Aware Cutout augmentation method, and also leverages a two-step transfer learning approach to improve the model performance further. Besides two-stage detectors, one-stage methods, such as YOLO [50] and its variants, also have been adapted to the TD problem. Considering the difference between objects in natural images and tables in the image documents, YOLOv3-TD [75] proposes an anchor optimization strategy and two post-processing methods to adjust the detection method. The authors of YOLOv3-TD observe that the width of a table is usually larger than its height unless the table is large. Based on this observation, they propose a K-means based method to optimize anchors and obtain more "horizontal" anchors. Besides, they also erase the white space margin from predicted regions and filter the noisy page objects as the post-processing methods to improve the model performance further. Besides these one-stage and two-stage methods, transformer-based approaches such as DETR [26] and Deformable-Detr [61], also have been applied to the TD problem [7, 24]. There are many other studies discussing the TD problem, including DeepDeSRT [76], TableDet [8] and many others [77, 78]. All in all, these studies usually adopt popular object detection models to the TD problem and use some specifically designed methods to improve the model performance based on the characteristics of the TD problem. As discussed in Chapter 1, it is critical to avoid information loss for the TD task, and the tables in the documents are often sparsely distributed and well-formatted. Therefore, applying detection models with sparse region proposals might be a better option than existing studies relying on one-stage models or two-stage models with dense region proposals. The proposed solution in this

thesis is based on Sparse R-CNN, which will be discussed in Chapter 4.

2.4 Table Structure Recognition

There have been many studies [79–83] discussing the TSR problem in recent years. As mentioned in Chapter 1, we can roughly categorize these solutions into image-to-sequence, detection-based, and graph-based models. Image-to-sequence based models usually define the ground truth as structured sequences, such as HTML sequences, built on the transformer architecture [60], and follow an encoder-decoder architecture. For instance, TableMaster [17] is a typical image-to-sequence based model that can generate HTML sequences. More specifically, TableMaster follows the architecture of MASTER [84], which is originally designed for the scene text generation following the transformer architecture [60], and further improved the encoder part by introducing a Multi-Aspect Global Context Attention. Besides, TableMaster has two branches designed for the HTML sequence generation and bounding box regression. Similarly, MTL-TabNet [16] also follows the encoder-decoder architecture but contains three decoders for the cell box regression, cell content recognition, and HTML sequence generation, respectively. DRCC [20] argues that the error accumulation problem degrades the performance of image-to-sequence TSR models, especially when the input image is large. Therefore, DRCC proposes a two-step decoder architecture, which first decodes the input image into rows and then decodes the rows in cell sequences. VAST [85] pays more attention to the imprecise bounding boxes of table cells and proposes a Coordinate Sequence Decoder to improve the model's ability to generate accurate bounding boxes and introduces a visual-alignment loss to align the visual and structural information. To sum up, this type of method is usually based on the encoder-decoder architecture and can be trained end-to-end without using post-processing methods. Since the ground truth sequences used in image-to-sequence models usually contain information regarding spanning cells and header cells, these models can handle complex structures with spanning cells and identify header cells.

On the other hand, detection-based models usually define the problem as detecting different table components and applying a post-processing method to reconstruct table structures. DeepTabStR [22] proposes to detect columns and rows to obtain the table cells. However, DeepTabStR ignores the row/column span in the tables, which means that it cannot recover the hierarchical structures of tables. TableStrRec [86] extends the DeepTabStR, defining four types of table components: regular columns, irregular columns, regular rows, and irregular rows. Then, the spanning cells across multiple columns can be inferred from the difference between the regular and irregular columns when they are

overlapped, and the spanning cells across multiple rows can be inferred from regular and irregular rows similarly. PubTables1M [24] is another typical detection-based approach that defines six table components: table, column, row, spanning cell, Projected Row Header, and Column Header, in which Projected Row Header and Column Header are for the function analysis, and other components can be used to reconstruct the complex table structure. Among these formulations, only the problem formulation of PubTables1M can provide as much information as image-to-sequence models because it can provide header cell information and reconstruct the complex table structure. Besides, these detection-based models need an extra deterministic rule-based post-processing method to infer the table structure from detected table components, meaning they are not end-to-end.

At last, graph-based methods usually apply either detection or segmentation methods to obtain the locations of table cells and further build the relation among table cells. For instance, TGRNet [87] formulates the cell location detection and cell logical location prediction jointly in a multi-task architecture, which is modularized by a segmentation based method and graph convolutional network (GCN), respectively. Similarly, TSRNet [88] proposes a unified GNN-based approach modeling table detection and table structure recognition tasks together. More specifically, TSRNet also employs a semantic segmentation module to extract primitive regions, then applies k-nearest neighbors and line-of-sight neighbors to construct the graph and further classify the graph nodes and edges to filter the noise regions, merge, and build relations. In contrast, LGPMA [89] proposes a Local Pyramid Mask Alignment Module and Global Pyramid Mask Alignment Module to localize table cells, which are formulated as detection and segmentation problems and can be implemented by MaskR-CNN [57]. To construct the structure of the table, LGPMA further proposes a pipeline of cell matching, empty cell searching, and empty cell merging using the Maximum Clique Search algorithm and rule-based methods. Besides building graphs explicitly, some studies [90–92] predict the table grids or separators first and then merge grid elements, which are also treating grid elements as graph nodes. SPLERGE [90] is a typical method following this strategy consisting of a Split Model and Merge Model, in which the Split Model consists of a Row Projection Network and a Columns Projection Network to obtain the table grid, and the Merge Model is used to merge the grid cells. Similarly, SEM [91] employs a segmentation model to segment columns and rows and generate the table grid with a post-processing method. After the table grid is obtained, SEM introduces an Embedder network to extract and fuse the features from textual and visual modalities. A Merger network takes the fused features from Embedder as inputs to merge the grid elements. TSRFormer-DQ-DETR [93] leverages a DETR [26] based separation line prediction model, termed DQ-DETR, to predict the reference points on separation lines, followed by a Relation Network based cell Merging module to merge grid elements.

RobustTabNet [94] employs a spatial network to predict Row and Column separation lines and further introduces a Grid CNN module to merge and build relations of table cells. Since these graph-based models identify the graph nodes first, defining a cell-type classification task is necessary if they want to provide information regarding header cells.

This thesis focuses on the well-formatted tables suitable for detection-based approaches. However, as discussed in Chapter 1, current studies usually have improper problem formulations and lack explorations of the underlying reasons for different design aspects. Therefore, considering that the defined columns and rows are densely and closely distributed in tables, this thesis proposes a Cascade R-CNN based approach, explores and explains the critical factors in the model design, including the problem formulation, regional proposal generation, feature extraction, and the alignment of optimization target with TSR metrics, which will be discussed in detail in Chapter 5.

2.5 Pre-trained Language Model

As mentioned in Chapter 1, since there is no formal definition for large language models, we term language models whose number of parameters is smaller than 1 billion as Pre-trained Transformer-based Models in this thesis. There have been many Pre-trained Transformer-based models for the text, image, and multi-modal tasks. For these models with text inputs, their model architectures usually rely on the Transformer Encoder, Decoder, or both Encoder and Decoder. For example, BERT [95] is a typical pre-trained model built on the Transformer Encoder architecture with text inputs. During training, BERT is trained with two self-supervised tasks: Masked LM [95] and Next Sentence Prediction [95]. RoBERTa [96] follows the architecture of BERT and further optimizes BERT in the training procedure, such as increasing the batch size, using large BPE vocabulary, and achieving more robust and promising results on the downstream tasks. ALBERT [97] also follows the architecture of BERT, introduces parameter reduction techniques to increase the training speed, and proposes a new loss to model inter-sentence coherence, resulting in better performance on downstream tasks. Instead of relying on Transformer Encoder, GPT [98] and its successors are built with Transformer Decoder architecture. GPT uses the self-supervised task of predicting the next word in a sequence based on the previously known words, often an auto-regressive pre-train task. Besides the models built with Transformer Encoder or Decoder only, BART [99] uses the Encoder-Decoder architecture combining Bidirectional and Auto-Regressive Transformers, applies five self-supervised pre-training tasks, including Token Masking, Token Deletion, Text Infilling, Sentence Permutation and Document Rotation [99]. It is worth mentioning that there are a lot of other Pre-

trained Transformer-based Models [100, 101], and these models can be applied to analyze the tabular data once the tabular data is transformed into text sequences [102]. Besides these models trained with large-scale text corpus, some studies further optimize them with large-scale tabular data for tabular understanding tasks. For example, TaPas [103] collects 6.2M tables from Wikipedia to further optimize BERT for tabular tasks. Based on TaPas, Omnitab [104] further proposes to use synthetic tabular data to pre-train the model for the Table-QA problem in the few-shot setting. Even though these Pre-trained Language Models can achieve promising results on the Table-QA tasks, they need to be fine-tuned on the downstream tasks and can only deal with text inputs.

On the other hand, many studies have proposed Pre-trained Transformer-based Models with image and multi-modal inputs. We only discuss document understanding models that are related to our Table-VQA problem. LayoutLM [105] is a typical multi-modal model following BERT architecture using both layout and image embeddings as inputs. During the fine-tuning stage of LayoutLM, both layout and image embeddings are used for the downstream tasks. Similarly, LayoutLMv2 [106] further extends LayoutLM, considering the layout, text, and image information of visually rich document images. It uses Masked Visual-Language Modeling, Text-Image Alignment, and Text-Image Matching as pre-trained tasks. There are many similar studies, such as Donut [107], Ernie-layout [108]. Since these models all use full document images as a training corpus, they are not optimized for the table images, making them lack the capacity to understand complex table structures and reason over table images.

2.6 Large Language Model

In this section, we briefly review the large language models (LLMs) whose number of parameters is larger than 1 billion. Overall, we can also categorize these models into two groups based on whether they can process multi-modal inputs: text-only and multi-modal LLMs. A LLM trained from scratch usually contains multiple training stages, including Pre-training, supervised fine-tuning and iterative Reinforcement Learning with Human Feedback (RLHF) stages [109]. GPT3.5 [110] is the breakthrough LLM, which has demonstrated remarkable performance on a variety of NLP tasks, but it is a closed source. After GPT3.5, a variety of open source LLMs released using similar pre-training techniques, such as Llama2 [109], Falcon [111], Mixtral-8*7B [112] and many others. Besides models trained from scratch, there are many models fine-tuned on the open-source LLMs. Alpaca [113] and Vicuna [114] are two popular models fine-tuned based on Llama [115]. More specifically, Alpaca utilizes 52K instruction-following data generated by self-instruct method [116]

to fine-tune the Llama, which achieves similar performance with GPT-3.5. Similarly, Vicuna [114] uses 70K samples collected from ShareGPT to fine-tune the Llama, achieving more than 90% quality of GPT3.5 based on the evaluation of GPT-4 [117].

Along with the fast development of text-only LLMs, multi-modal LLMs, which can take images as inputs, are also developing fast and have shown promising performance. GPT-4V(vision) [117, 118] is the leading multi-modal LLM, but it is closed-sourced, and its implementation details are unknown. For the open source alternatives, they often freeze or use parameter-efficient methods to fine-tune the text-only LLM and the Pre-trained Vision Model and align the vision embedding with text embedding by adding extra layers and fine-tuning with new data, such as Blip-2 [119], MiniGPT4 [120], MiniGPT-v2 [121], Flamingo [122]. Even though these multi-modal models have shown promising potential, there are still many challenges, such as the limited image resolution and the hallucinations. To increase the resolution of input images, Monkey [123] proposes to divide an image with a high resolution into uniform patches and process the patches in parallel with individual adapters to obtain the local features. Besides, the original image is also resized to the size of each patch to obtain the feature, and the local features and the global features are fused by a Shared Resampler. Monkey is trained with public document datasets and can show promising results on various image-based Document QA datasets. LLaVA-UHD [124] is another study for high-resolution images and further extends it to deal with various aspect ratios. More specifically, LLaVA-UHD divides original images into smaller variable-sized instead of fixed-size patches in Monkey [123] and uses a compression layer to compress the length of tokens to reduce the computations. Besides, the spatial information is fed to the LLM together with the visual tokens in LLaVA-UHD to improve its performance further. Similar to the text-only LLMs, some studies [125,126] also apply RLHF to the multi-modal models to alleviate the hallucinations. For example, RLAIF-V [125] proposes a candidate response generation strategy to generate high-quality data pairs and apply an iterative alignment framework to mitigate the distribution shift. The directions discussed in this section are developing fast, and many studies have discussed these topics. Since these discussed methods usually have high computation resources and dataset requirements, this thesis focuses on the training-free prompting engineering methods for the Table-QA problem.

2.7 Table Question Answering

2.7.1 LLM-based Methods for Table Question Answering

Since this study focuses on applying LLM to the Table-QA problem, we only include recent studies using LLM to solve the Table-QA problem in this section. Specifically, most of the studies applying LLMs to the Table-QA task can be categorized into instruction tuning based and prompt engineering based approaches. Instruction-tuning approaches usually need to collect large-scale datasets and then further fine-tuned LLMs with parameter-efficient fine-tuning methods, such as LORA [127] and LongLORA [128]. TableLLAMA [129] and TAT-LLM [130] are typical examples of applying instruction tuning for the table processing. TableLLAMA is a generalist model for tables fine-tuned on Llama2 [109] with LongLORA. For fine-tuning TableLLAMA, a dataset collection named TableInstruct is proposed by collecting table-based samples from 14 datasets for 11 tasks. TAT-LLM is another fine-tuned LLAMA2 model specifically for the discrete reasoning over tables, which decomposes the Table-QA into three steps: Extractor, Reasoner and Executor. To construct the dataset for the model fine-tuning, TAT-LLM proposes a template to guide LLM in generating data following the proposed step-wise pipeline.

On the other hand, prompting engineering based solutions focus on proposing proper prompts to the language model to elicit LLMs' capacities. Since Table-QA needs to extract relevant information and evidence from tables and perform reasoning, decomposing the Table-QA task into multiple steps is also widely adopted in prompting engineering-based solutions. Besides, as LLM often fails to process large tables and complex reasoning, many studies [32, 38, 130, 131] propose to decompose large tables into small tables in the extraction step and divide the complex reasoning task into more straightforward reasoning questions. For example, StructGPT [132] defines specialized interfaces for information extraction and linearizes the extracted sub-tables as inputs to the LLM for question answering. EEDP [133] is another example proposing a prompt method containing four steps: Elicit, Extract, Decompose and Predict.

Besides these studies decomposing the Table-QA into extraction and reasoning steps and dividing difficult extraction and reasoning tasks into simpler ones, programming languages, such as SQL and Python, are also widely leveraged in these solutions to overcome the limitations of LLMs in arithmetic calculation and reasoning. Dater [38] is a typical solution following the multi-step design and leveraging SQL to conduct reasoning. More specifically, Dater decomposes both large evidence and complex questions into relevant and simpler ones, applies SQL queries to produce numerical and logical reasoning results to the decomposed sub-questions, and generates the final results based on the sub-results and

extracted evidence with ICL. Binding [37] is another example of applying Python and SQL for table processing, which proposes a unified API to map tasks into executable programs.

All in all, as the complexities of Table-QA in the information extraction and reasoning, breaking Table-QA into multiple steps and decomposing difficult extraction and reasoning tasks into simpler ones have been the dominant solution, and external tools such as SQL and Python can compensate for the limitations of LLMs in arithmetic calculation and reasoning. Therefore, considering the limitations of LLMs, this thesis also employs Python code for the reasoning steps. Besides, as the generated Python often fail to execute because of the heterogeneous data types of tables and complex table structures, this thesis proposes a three-stage prompting approach, including planning, reasoning and correction. More details of the proposed solution will be discussed in Chapter 6.

2.7.2 Table Question Answering Datasets

There have been many studies [36, 102, 134, 135] proposing datasets and solutions for the Table-QA problem. These datasets often use Wikipedia as the data source and generate questions and answers through crowdsourcing. WikiTableQuestions [36] is a popular Table-QA dataset using tables from Wikipedia and annotated by crowdsourcing. Two crowdsourcing tasks were created, the first to develop the questions based on the tables and 36 generic prompts and the second to answer the questions from task one. SequentialQA [134] further extended the WikiTableQuestions dataset, decomposing complex questions into a series of simple, interrelated question sequences so that the dataset can be used to explore the conversational QA setting. Different from WikiTableQuestions and SequentialQA, using short-form entities as Answers, FetaQA [135] uses free-form text as Answers, resulting in many more words in Answers. Besides these datasets focusing on the information entirely from the tables, HYBRIDQA [102] develops the question in a heterogeneous setting, meaning the information should be from heterogeneous sources, such as text passages and tables. Besides these datasets using short-term entities and free-term text as Answers, the Fact Verification task can be a variation of a typical Table-QA problem, which usually uses Evidence and Claim pairs as inputs and gives the result of whether the Evidence supports, refutes, or cannot verify the Claim. Similar to the aforementioned datasets, Fact Verification datasets also consider text and table information separately or jointly. For example, SEM-TAB-FACTS [136] is a dataset focusing on tabular data in scientific documents using generated tables. TabFact [137] is another dataset that focuses on the tables but collects them from Wikipedia. FEVEROUS [138] is another Fact Verification dataset using Wikipedia as the data source but considers both text and table sources. Different from the datasets mentioned above, GeoTSQA [139] is a special dataset created

by collecting multiple choices questions in the geography domain from China's high-school exams given tables as context information, meaning that the answers are one choice in four choices for each question. Studies proposing these datasets also include their solutions for the Table-QA problem, mainly using text-only pre-trained models, such as BERT [95] and their variations.

There are also a few multi-modal datasets for the QA problem. For example, MMQA [140] collects tables, attaching images, and text paragraphs to create the dataset, requiring the capacity of reasoning over multiple modalities and using short-from entities as answers. VQAonBD [141] merely focuses on the table images from the business documents with questions in text format and also uses short-form entities as answers, which means that it is also a multi-modal dataset. We summarize the key aspects of these datasets in Table 2.1. Since WikiTableQuestions and TabFact are the two most popular datasets in Table-QA studies, we use these two datasets to verify the proposed Table-QA solution in this thesis.

Table 2.1: Summary of Table-QA and Table-VQA datasets.

Datasets	Modality	Data Source	Answer Format	Avg # Words in Answer
FetaQA [135]	Text	Wikipedia	Free-form Text	18.9
WikiTableQuestions [36]	Text	Wikipedia	Short-form Entity	1.7
SequentialQA [134]	Text	Wikipedia	Short-form Entity	1.2
GeoTSQA [139]	Text	High-school Exams	Answer Choice	1.0
HYBRIDQA [102]	Text	Wikipedia	Short-form Entity	2.1
FEVEROUS [138]	Text	Wikipedia	Verification Result	1.0
SEM-TAB-FACTS [136]	Text	Scientific Documents	Verification Result	1.0
TabFact [137]	Text	Wikipedia	Verification Result	1.0
MMQA [140]	Image & Text	Wikipedia	Short-form Entity	2.1
VQAonBD [141]	Image & Text	Business Documents	Short-form Entity	1.0

2.8 Prompting Methods

This section focuses on recent prompting methods for LLMs. As some of these methods have also been applied to the Table-QA problem, some methods included in this section can be overlapped with the discussed studies in Section 2.7. There have been many studies demonstrated that prompting LLMs with a few demonstrations and intermediate reasoning steps can elicit the reasoning capacities of LLMs, which often refer to In Context Learning (ICL) [47] and CoT [31]. Although ICL and CoT are useful, writing proper prompting demonstrations is non-trivial and time-consuming. Therefore, some studies [142–144] propose to use LLMs to generate demonstrations. Auto-CoT [142] proposes to prompt LLMs

with Zero-shot CoT to generate reasoning rationales but finds that the generated rationales often contain mistakes. To mitigate the wrong demonstration issue, Auto-CoT proposes clustering and sampling the questions first and then applying simple heuristics to sample simpler questions and rationales to construct demonstrations. Synthetic Prompting [143] is another typical solution for constructing demonstrations with LLMs, containing forward and backward processes. In the backward process of Synthetic Prompting, a topic word, a target complexity and a self-generated reasoning chain are used as conditions to generate the synthetic question, and the synthetic question is used in the forward process to generate the precise synthetic reasoning chain. Along with these studies focusing on demonstration generation with LLMs, ensemble methods are also effective for reasoning. For example, self-consistency decoding [41] first generates a set of candidate outputs from the LLM with different sampling methods, such as temperature sampling, and then aggregates the sampled outputs and uses the most consistent output as the final result. Multi-Chain Reasoning [145] is another ensemble method mixing information from multiple reasoning chains. Different from studies [41] to ensemble results of multi-reasoning chains, Multi-Chain Reasoning collects evidence from multiple reasoning chains and prompts the LLM to give the final answer.

In addition to these prompting methods, the integration of programming languages and other external tools holds great potential for overcoming the limitations of LLMs. For instance, LLMs often struggle with precise arithmetic calculations, especially division operations, and staying updated with the latest information. To address these issues, PoT [30], PAL [29], and many other studies [146] leverage Programming Language to assist the reasoning steps. Some studies [147] introduce the concept of Agent and call external tools by parsing the LLM outputs to enhance the LLM's capabilities. These advancements in our field inspire optimism about the future of LLM research and its potential for further improvement.

Chapter 3

Revisiting Table Detection Datasets

3.1 Motivation

As discussed in Chapter 1, TD is the first step of the proposed pipeline in this thesis. There have been some public datasets for the table detection problem, such as ICDAR2013 [1], ICDAR2017 [2], ICDAR2019 [3] and TNCR [7]. However, these open-source datasets are suffering from the following issues. First, the labelling definitions across these datasets are different, making it difficult to merge smaller ones into large datasets. Second, the data sources of these datasets are very limited, primarily from academic publications or public governmental documents, which means that these datasets also cannot reflect the complexity of real enterprise applications. Third, these open-source datasets are often noisy, especially the ones generated by parsing metadata of documents. Therefore, this thesis proposes two new table detection datasets, Open-Tables and ICT-TD. The former dataset is constructed by cleaning data noise, aligning labelling definitions and merging several small open datasets. The latter dataset is constructed with datasheets from the ICT domain, containing various samples that hardly appear in the open-source datasets. For example, Figure 3.3 (1) is a special table containing several sub-tables, and Figure 3.3 (7) is a table containing figures as the content of some table cells.

To sum up, this chapter covers the following aspects of the proposed datasets.

1. This chapter revisits several open-source TD datasets, aligns their annotations, cleans the noisy labels, and merges them to form a new Open-Tables dataset. The Open-Tables dataset can provide a more reliable model evaluation, minimizing the side effects of noisy labels.

2. A new manually annotated dataset, ICT-TD, is proposed using the PDF files of ICT commodities containing many domain-specific training samples. The ICT-TD dataset provides a new data source with many unique samples that can hardly be found in other open-source datasets.

3. In addition to various strong baselines using different types of state-of-the-art object detection methods, this chapter also presents benchmarks in the cross-domain setting.

The rest of this chapter is organized as follows: Sections 3.2 and 3.3 introduce the proposed Open-Tables and ICT-TD datasets. Section 3.4 builds the baselines, analyzes the characteristics of the proposed datasets, and discusses the potential applications. At last, Section 3.5 summarizes this chapter and discusses possible research directions regarding the TD datasets.

3.2 Open-Tables Dataset

As discussed in Section 2.2, ICDAR2019 contains archival and modern documents. This thesis only uses its modern documents. Since there are five find-grained types of tables in the TNCR dataset, all these annotations are transformed into a single type, namely tables. Even though the annotation quality of the datasets used here is relatively higher, many samples still have noisy annotations that have the issues shown in Figure 1.2. These samples in the test set can influence the model evaluation, and noisy samples in the training set can degrade the model performance. Therefore, these noisy samples are first corrected.

Besides the noisy annotations, as discussed in Chapter 1 and 2.2, the table definition across these open-source datasets can differ. This issue is caused by ambiguous samples. Figure 3.1 shows two ambiguous samples from the TNCR dataset. The first ambiguous sample shows two alternative bounding boxes that can cause inconsistent annotation issues. The ground truth (green box) provided by the TNCR dataset excludes the explanation part of the table. The second ambiguous sample is labeled as a table but is unnecessary in other datasets because it can also be defined as a document footer. To address these ambiguous samples, this thesis defines the following rules to align the datasets. First, the thesis uses the table lines as the priority, meaning that all the content bounded by the table lines should be included. However, when table lines do not bind a table explanation part, it should not be defined as part of the table. Second, a table should at least have two lines and two columns. Following these two rules, the explanation part of the first sample should be included, and the second sample should be defined as none-table, as shown in Figure 3.1.

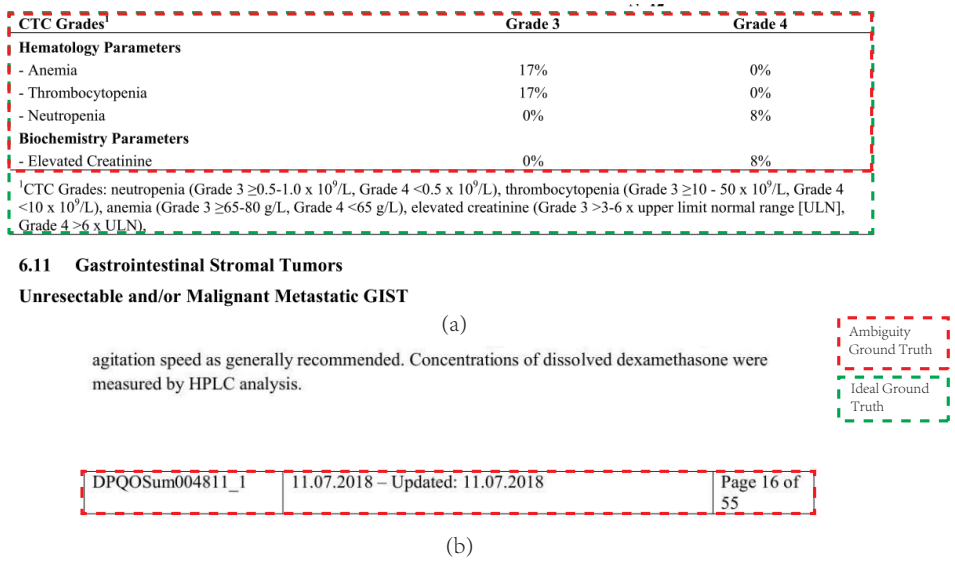


Figure 3.1: Three ambiguity samples. Figure (a) shows two alternative annotations of a table. The green bounding box in Figure (a) is the ground truth provided by the TNCR dataset, which excludes the table explanation part. The red bounding boxes in Figure (b) are defined ground truth but are not tables in other datasets.

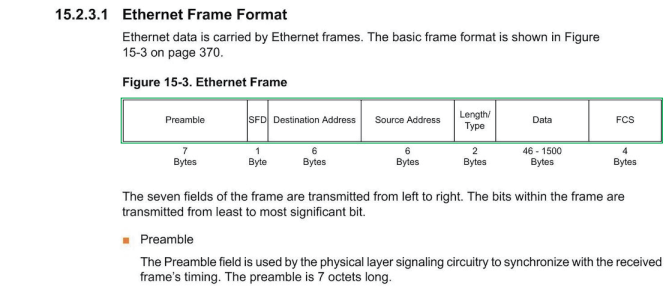
After data cleaning and ambiguity resolving, the training sets of ICDAR2017, ICDAR2019, Marmot, and TNCR datasets are merged as the training set of the Open-Tables dataset, and the samples from the test sets of ICDAR2013, ICDAR2017, ICDAR2019, and TNCR datasets are merged as the test set. At last, the TNCR's validation set is used as the Open-Tables dataset's validation set. Since the Open-Tables dataset is an ensemble of public datasets, the statistics of these datasets are summarized in Tables 3.1 and 3.2 regarding the number of tables in training, testing, and validation sets, average image, table sizes, data sources and others. It is worth mentioning that there is no small object whose size is less than 32 * 32 in these datasets following the definition of COCO metric [28].

Table 3.1: Statistics of the datasets. The numbers reported in the columns of Training, Validation, and Testing denote the number of tables.

Dataset	Training	Testing	Validation	Avg. Image Size	Avg. Table Size
ICDAR2013	-	156	-	2,170 * 1,626	648 * 1172
ICDAR2017	713	321	-	1,081 * 802	198 * 421
ICDAR2019	981	449	-	1,078 * 812	261 * 582
Marmot	1,436	-	-	1,120 * 800	282 * 516
TNCR	6,384	1,346	1,427	1,548 * 1,277	439 * 917
Open-Tables	9,537	2,272	-	1,378 * 1,092	388 * 805
ICT-TD	6,082	1,700	1,427	1,963 * 1,487	452 * 987

3.3 ICT-TD Dataset

To create the ICT-TD dataset, 175,682 PDF documents for 370 different ICT commodities are harvested from the internet. Since each PDF file may have more than one page, each page is transformed into an image with a resolution of 200 DPI, resulting in 3,581,805 images. Then, a random sampling method is employed to select 5,000 samples from these images and manually annotate the bounding boxes of all the tables in the images. It is worth mentioning that all these PDF documents are well-formatted. Therefore, there are no distorted tables in the ICT-TD dataset. The statistics of the ICT-TD dataset and some public datasets are summarized in Tables 3.1 and 3.2 for comparison purposes. ICDAR2013 [1] is a small dataset without providing a training set. ICDAR2017 [2], ICDAR2019 [3], Marmot [6], and TableBank [4] are all using academic publications or public governmental documents as the data sources, making these datasets cannot reflect the complexity of real enterprise cases and hard to be adapted to the ICT domain.



15.2.3.1 Ethernet Frame Format

Ethernet data is carried by Ethernet frames. The basic frame format is shown in Figure 15-3 on page 370.

Figure 15-3. Ethernet Frame

Preamble	SFD	Destination Address	Source Address	Length/ Type	Data	FCS
7 Bytes	1 Byte	6 Bytes	6 Bytes	2 Bytes	46 - 1500 Bytes	4 Bytes

The seven fields of the frame are transmitted from left to right. The bits within the frame are transmitted from least to most significant bit.

- Preamble

  The Preamble field is used by the physical layer signaling circuitry to synchronize with the received frame's timing. The preamble is 7 octets long.

Figure 3.2: A sample of a single row figure that is not annotated as table. The single-row figure is highlighted with a green box.

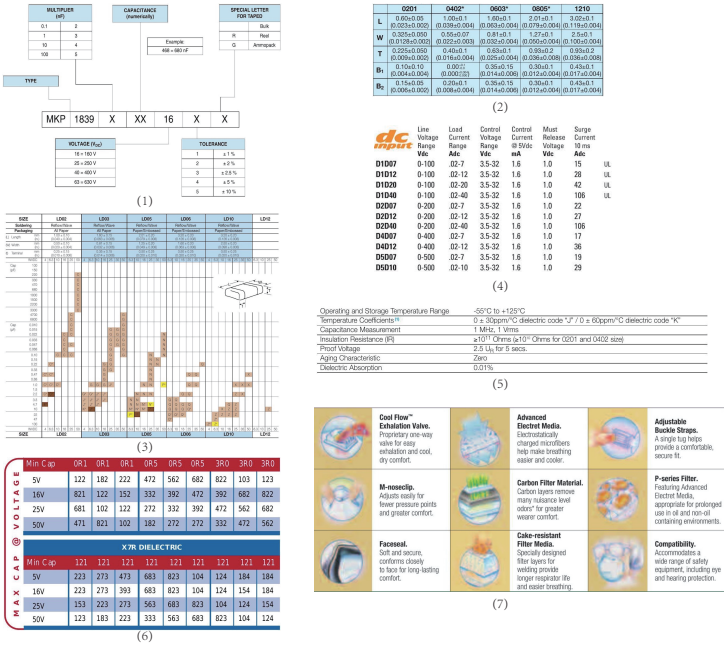
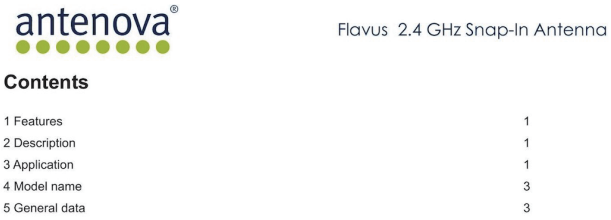


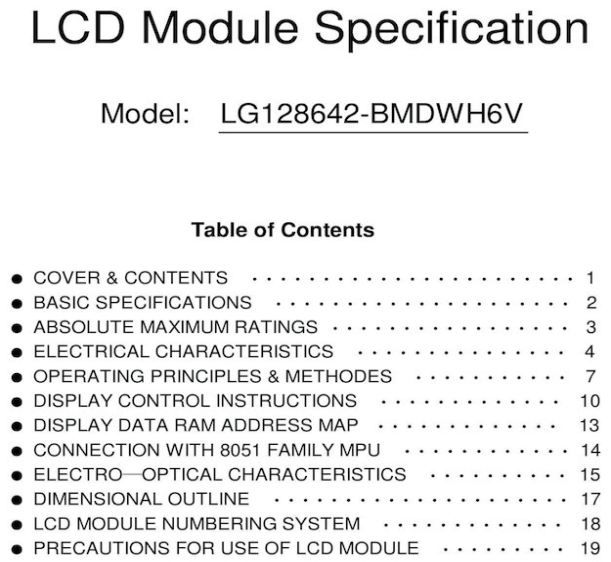
Figure 3.3: Table examples from the proposed dataset.


antenova®

Flavus 2.4 GHz Snap-In Antenna

Contents

(a) A sample whose appearance looks like a non-lined table but is not annotated as a table following our defined annotations rules.


LCD Module Specification

Model: LG128642-BMDWH6V

Table of Contents

(b) A sample which is described as a table but is not annotated as a table following our defined annotations rules.

Figure 3.4: Two samples are not annotated as tables because they do not describe ICT commodities and are not useful for downstream tasks.

Since many ambiguous cases exist in the ICT domain documents, the following rules are defined to annotate tables. First, a table must at least have two rows and two columns because a table should be a summary of critical information. The ones with a single row or column are treated as plain text or figures. Figure 3.2 shows an example of a single-row figure whose appearance resembles a table but should not be annotated as a table following this rule. Instead, it should be a figure. Second, tables should contain information describing the commodities because this information is useful for domain-specific applications. Some information can be formatted like tables, such as the index page of the document, but not useful for the downstream tasks. Figure 3.4 shows two samples that are not annotated as tables because they are the index of content without containing information on any commodities. However, their appearances are similar to tables, making this dataset more challenging. Third, table titles and table notes should not be included in the tables unless there are lines to have them as parts of a table because we focus on the TD problem. Table titles and table notes should be treated as different components in a document, which is beyond the scope of this study.

Following these rules, tables in the proposed ICT-TD dataset can be grouped into four categories based on the content and the structure of tables: fully-lined tables, partially-lined tables, non-lined tables, and other unique tables. Figure 3.3 shows some samples of these different types of tables in the proposed ICT-TD dataset. Figure 3.3 (a) is a special table comprising many sub-tables. Since each of these sub-tables describes a parameter of the commodity, the union of these sub-tables is treated as a single special table. Figures 3.3 (d) (e) are two examples of non-lined and partially-lined tables, respectively. Figures 3.3 (b) (c) (f) (g) are fully-lined tables with unique appearances. For example, Figure 3.3 (f) is a combination of two sub-tables. Figure 3.3 (g) uses small figures as the content of table cells. These tables are challenging for the models trained with open-source datasets because open-source datasets can hardly contain similar samples.

3.4 Experiments Results and Analysis

3.4.1 Main Results

This section presents the experiments and baselines for the proposed datasets. Four state-of-the-art approaches are selected as baseline models, including TableDet [8], Diffusion-Det [13], Deformable-DETR [61] and Sparse R-CNN [10]. TableDet is built on Cascade-RCNN [58] leveraging transfer learning and table-aware data augmentation to improve the performance for the TD problem further. Deformable-DETR is a typical transformer-based

Table 3.2: Statistics of the ICT-TD dataset and public datasets. Notably, the values in this table are the number of images, not the number of tables. * means the datasets are used to create the Open-Tables dataset.

Dataset	Type	Training	Testing	Validation	Data Source
TableBank	Generated	130,463	5,000	10,000	Word and Latex
PubLayNet	Generated	86,950	3,772	3,950	PubMed
ICDAR2013*	Manual	–	238	–	Governmental Documents
ICDAR2017*	Manual	1,600	817	–	CiteSeer
ICDAR2019*	Manual	600	240	–	Journals, Financial STMT
Marmot*	Manual	2,000	–	–	E-book and CiteSeer
TNCR*	Manual	4,634	1000	1,015	Governmental Documents
IIIT-AR-13K	Manual	9,000	2,000	2,000	Annual Reports
Open-Tables	Manual	88,34	2,295	1,015	Merged dataset
ICT-TD	Manual	4,000	1,000	–	ICT PDF Documents

approach, and DiffusionDet introduces diffusion process [148, 149] to the object detection problem with random region proposals. Sparse R-CNN is a typical method using learnable region proposals. Thus, baseline models contain a two-stage model, a transformer-based model, a model with random proposals, and a model with learnable region proposals to cover the most popular object detectors. It is worth mentioning that one-stage detectors are not included as baseline models because one-stage detectors are usually not as good as other types of detectors included here for the TD task. TableDet is re-implemented with Detectron2 [150], keeping the table-aware augmentation method. The implementation of Deformable-DETR can be found in detrex [151]. DiffusionDet and Sparse R-CNN have their official implementations. All these baseline models use ResNet50 [152], pre-trained on ImageNet [153] as the training start point. Notably, the original design of TableDet uses pre-trained CascadeMaskR-CNN on COCO dataset [28] as the initialization. The default model parameter configurations of these benchmark models are used, but some parameters are tuned regarding the training scheduling of DiffusionDet, Deformable-DETR, and Sparse R-CNN because their default training scheduling parameters are tuned based on the COCO dataset. Some key parameters are summarized in Table 3.3. It is worth mentioning that since all these benchmark models are built on Detection2, terms in Table 3.3 are also following Detectron2. For the training of all models, we use the validation set for the model selection and hyper-parameter tuning.

Precision, recall, and F1-score are employed as the evaluation metrics. An IoU score is used as the threshold to determine whether a table is detected, which can be calculated

Table 3.3: Key parameters of the benchmark models.

Method	TableDet	DiffusionDet	Deformable-DETR	Sparse R-CNN
OPTIMIZER	SGD	AdamW	AdamW	AdamW
MAX_ITER	25,000	50,000	50,000	50,000
MAX_EPOCH	100	200	200	200
STEPS	-	37,500	37,500	37,500
SCHEDULER	-	MultiStepLR	MultiStepLR	MultiStepLR
BASE_LR	1.0e-03	1.0e-05	1.0e-04	2.5e-05
GAMMA	-	0.1	0.1	0.1
IMS_PER_BATCH	16	16	16	16

by Equation 3.1. Then, the True Positive is the number of predictions whose IoU scores to one of the ground truth bounding boxes are larger than an IoU threshold, and these corresponding ground truth bounding boxes are treated as being detected. Similarly, the False Positive can be calculated as the number of predictions whose IoU to all ground truths bounding boxes that are less than the IoU threshold, and the False Negative is the number of ground truth bounding boxes that are not detected. At last, the Precision, Recall and F1-score can be calculated by Equation 3.2, 3.3 and 3.4, respectively.

As mentioned in Chapter 1, the TD problem requires the detectors to maintain adequate precision and recall when the IoU threshold is high, and scores with larger IoU thresholds are more discriminate. Therefore, this thesis follows the ICDAR2019 competition [3] to use weighted F1-score as the primary evaluation metric, defined as Equation 3.5. 80%, 85%, 90%, and 95% are chosen as the IoU thresholds instead of 60%, 70%, 80%, and 90% used in the ICDAR2019 competition.

Figures 3.5 and 3.6 present some prediction samples of the baseline models. Sub-Figures (a) (b) (c) (d) (e) (f) are the original image, the ground truth, and the results of TableDet, DiffusionDet, Deformable-DETR, and Sparse R-CNN, respectively. For the table in Figure 3.5, its ground truth should contain the table explanation part because a bottom line bounds the explanation texts. However, the prediction box of TableDet is not large enough to cover all the explanation texts. Deformable-DETR does not treat explanation texts as part of the table, but its prediction box can fit other parts well. By contrast, DiffusionDet and Sparse R-CNN can detect this table very well. For the table in Figure 3.6, TableDet and DiffusionDet can detect two tables successfully, even though their prediction boxes cannot fit the table precisely. By contrast, Deformable-DETR detects two tables as a single table, and Sparse R-CNN missed the second table at the bottom. These

Table 3.4: Experimental results on the ICT-TD dataset with F1-score. 4000 and 1000 are the number of samples.

Dataset		Model	F1 under IoU thresholds				WAvg. F1
Training	Testing		80%	85%	90%	95%	
ICT-TD Training Set (4000)	ICT-TD Test set (1000)	TableDet	93.9	92.4	89.6	75.9	87.5
		DiffusionDet	95.5	94.2	91.3	76.5	88.9
		Deformable-DETR	95.1	93.8	91.6	82.1	90.3
		Sparse R-CNN	94.3	92.9	90.4	79.3	88.9

samples show that the baseline models have different weaknesses in detecting tables from the proposed Open-Tables dataset, demonstrating that the Open-Tables dataset can be a useful source for TD studies. Similarly, several prediction samples on the ICT-TD dataset are included in Figures 3.7 and 3.8. The ideal boxes of tables in Figure 3.7 should cover their table header cells that are not bounded by lines. However, only TabeDet can cover these header cells, but it detects the upper table as two tables. Figure 3.8 shows a sample where all four baseline models recognize a figure as a table. It is worth mentioning that samples in Figures 3.7 and 3.8 are domain-specific, making the ICT-TD dataset a useful source for ICT domain and cross-domain applications.

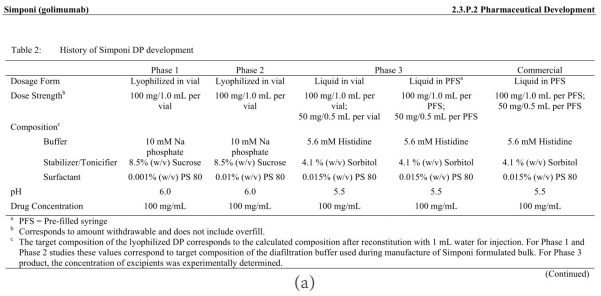
$$IoU = \frac{Overlap\ Area\ of\ two\ Boxes}{Union\ Area\ of\ two\ Boxes} \tag{3.1}$$

$$\text{Precision} = \frac{\text{True positive}}{\text{True positive} + \text{False positive}} \tag{3.2}$$

$$\text{Recall} = \frac{\text{True positive}}{\text{True positive} + \text{False negative}} \tag{3.3}$$

$$\text{F1-score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.4}$$

$$\text{Weighted Avg. F1-score} = \frac{\sum_{i=1}^{4} IoU_i \cdot F1@IoU_i}{\sum_{i=1}^{4} IoU_i} \tag{3.5}$$



Simponi (golimumab) 2.3.P.2 Pharmaceutical Development

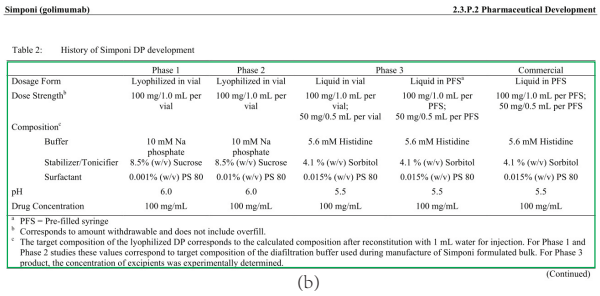
Table 2: History of Simponi DP development

	Phase 1	Phase 2	Phase 3		Commercial
Dosage Form	Lyophilized in vial	Lyophilized in vial	Liquid in vial	Liquid in PFS[a]	Liquid in PFS
Dose Strength[b]	100 mg/1.0 mL per vial	100 mg/1.0 mL per vial	100 mg/1.0 mL per vial; 50 mg/0.5 mL per vial	100 mg/1.0 mL per PFS; 50 mg/0.5 mL per PFS	100 mg/1.0 mL per PFS; 50 mg/0.5 mL per PFS
Composition[c]					
Buffer	10 mM Na phosphate	10 mM Na phosphate	5.6 mM Histidine	5.6 mM Histidine	5.6 mM Histidine
Stabilizer/Tonicifier	8.5% (w/v) Sucrose	8.5% (w/v) Sucrose	4.1 % (w/v) Sorbitol	4.1 % (w/v) Sorbitol	4.1 % (w/v) Sorbitol
Surfactant	0.001% (w/v) PS 80	0.01% (w/v) PS 80	0.015% (w/v) PS 80	0.015% (w/v) PS 80	0.015% (w/v) PS 80
pH	6.0	6.0	5.5	5.5	5.5
Drug Concentration	100 mg/mL	100 mg/mL	100 mg/mL	100 mg/mL	100 mg/mL

[a] PFS = Pre-filled syringe
[b] Corresponds to amount withdrawable and does not include overfill.
[c] The target composition of the lyophilized DP corresponds to the calculated composition after reconstitution with 1 mL water for injection. For Phase 1 and Phase 2 studies these values correspond to target composition of the diafiltration buffer used during manufacture of Simponi formulated bulk. For Phase 3 product, the concentration of excipients was experimentally determined.

(Continued)

(a)



Simponi (golimumab) 2.3.P.2 Pharmaceutical Development

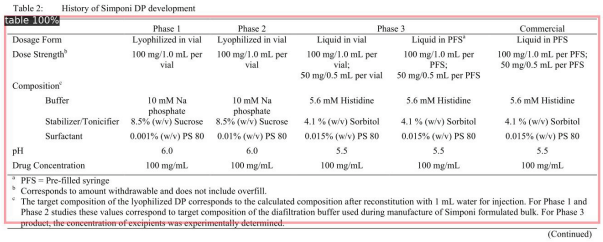
Table 2: History of Simponi DP development

	Phase 1	Phase 2	Phase 3		Commercial
Dosage Form	Lyophilized in vial	Lyophilized in vial	Liquid in vial	Liquid in PFS[a]	Liquid in PFS
Dose Strength[b]	100 mg/1.0 mL per vial	100 mg/1.0 mL per vial	100 mg/1.0 mL per vial; 50 mg/0.5 mL per vial	100 mg/1.0 mL per PFS; 50 mg/0.5 mL per PFS	100 mg/1.0 mL per PFS; 50 mg/0.5 mL per PFS
Composition[c]					
Buffer	10 mM Na phosphate	10 mM Na phosphate	5.6 mM Histidine	5.6 mM Histidine	5.6 mM Histidine
Stabilizer/Tonicifier	8.5% (w/v) Sucrose	8.5% (w/v) Sucrose	4.1 % (w/v) Sorbitol	4.1 % (w/v) Sorbitol	4.1 % (w/v) Sorbitol
Surfactant	0.001% (w/v) PS 80	0.01% (w/v) PS 80	0.015% (w/v) PS 80	0.015% (w/v) PS 80	0.015% (w/v) PS 80
pH	6.0	6.0	5.5	5.5	5.5
Drug Concentration	100 mg/mL	100 mg/mL	100 mg/mL	100 mg/mL	100 mg/mL

[a] PFS = Pre-filled syringe
[b] Corresponds to amount withdrawable and does not include overfill.
[c] The target composition of the lyophilized DP corresponds to the calculated composition after reconstitution with 1 mL water for injection. For Phase 1 and Phase 2 studies these values correspond to target composition of the diafiltration buffer used during manufacture of Simponi formulated bulk. For Phase 3 product, the concentration of excipients was experimentally determined.

(Continued)

(b)



Simponi (golimumab) 2.3.P.2 Pharmaceutical Development

Table 2: History of Simponi DP development

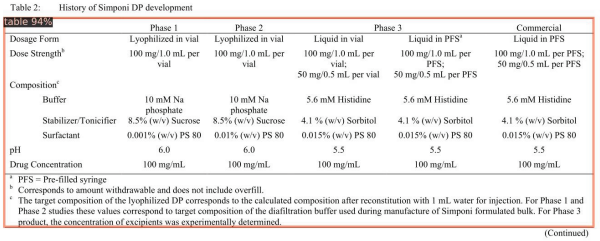
table 100%

	Phase 1	Phase 2	Phase 3		Commercial
Dosage Form	Lyophilized in vial	Lyophilized in vial	Liquid in vial	Liquid in PFS[a]	Liquid in PFS
Dose Strength[b]	100 mg/1.0 mL per vial	100 mg/1.0 mL per vial	100 mg/1.0 mL per vial; 50 mg/0.5 mL per vial	100 mg/1.0 mL per PFS; 50 mg/0.5 mL per PFS	100 mg/1.0 mL per PFS; 50 mg/0.5 mL per PFS
Composition[c]					
Buffer	10 mM Na phosphate	10 mM Na phosphate	5.6 mM Histidine	5.6 mM Histidine	5.6 mM Histidine
Stabilizer/Tonicifier	8.5% (w/v) Sucrose	8.5% (w/v) Sucrose	4.1 % (w/v) Sorbitol	4.1 % (w/v) Sorbitol	4.1 % (w/v) Sorbitol
Surfactant	0.001% (w/v) PS 80	0.01% (w/v) PS 80	0.015% (w/v) PS 80	0.015% (w/v) PS 80	0.015% (w/v) PS 80
pH	6.0	6.0	5.5	5.5	5.5
Drug Concentration	100 mg/mL	100 mg/mL	100 mg/mL	100 mg/mL	100 mg/mL

[a] PFS = Pre-filled syringe
[b] Corresponds to amount withdrawable and does not include overfill.
[c] The target composition of the lyophilized DP corresponds to the calculated composition after reconstitution with 1 mL water for injection. For Phase 1 and Phase 2 studies these values correspond to target composition of the diafiltration buffer used during manufacture of Simponi formulated bulk. For Phase 3 product, the concentration of excipients was experimentally determined.

(Continued)

(c)



Simponi (golimumab) 2.3.P.2 Pharmaceutical Development

Table 2: History of Simponi DP development

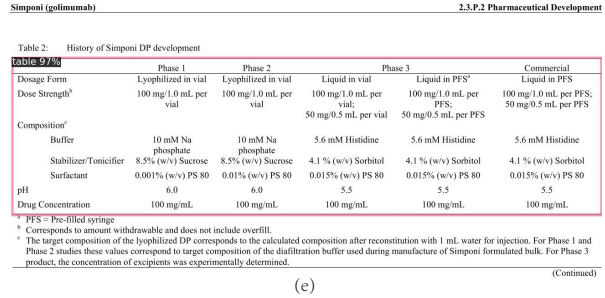
table 94%

	Phase 1	Phase 2	Phase 3		Commercial
Dosage Form	Lyophilized in vial	Lyophilized in vial	Liquid in vial	Liquid in PFS[a]	Liquid in PFS
Dose Strength[b]	100 mg/1.0 mL per vial	100 mg/1.0 mL per vial	100 mg/1.0 mL per vial; 50 mg/0.5 mL per vial	100 mg/1.0 mL per PFS; 50 mg/0.5 mL per PFS	100 mg/1.0 mL per PFS; 50 mg/0.5 mL per PFS
Composition[c]					
Buffer	10 mM Na phosphate	10 mM Na phosphate	5.6 mM Histidine	5.6 mM Histidine	5.6 mM Histidine
Stabilizer/Tonicifier	8.5% (w/v) Sucrose	8.5% (w/v) Sucrose	4.1 % (w/v) Sorbitol	4.1 % (w/v) Sorbitol	4.1 % (w/v) Sorbitol
Surfactant	0.001% (w/v) PS 80	0.01% (w/v) PS 80	0.015% (w/v) PS 80	0.015% (w/v) PS 80	0.015% (w/v) PS 80
pH	6.0	6.0	5.5	5.5	5.5
Drug Concentration	100 mg/mL	100 mg/mL	100 mg/mL	100 mg/mL	100 mg/mL

[a] PFS = Pre-filled syringe
[b] Corresponds to amount withdrawable and does not include overfill.
[c] The target composition of the lyophilized DP corresponds to the calculated composition after reconstitution with 1 mL water for injection. For Phase 1 and Phase 2 studies these values correspond to target composition of the diafiltration buffer used during manufacture of Simponi formulated bulk. For Phase 3 product, the concentration of excipients was experimentally determined.

(Continued)

(d)



Simponi (golimumab) 2.3.P.2 Pharmaceutical Development

Table 2: History of Simponi DP development

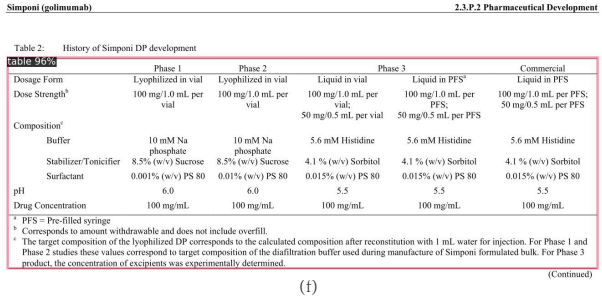
table 97%

	Phase 1	Phase 2	Phase 3		Commercial
Dosage Form	Lyophilized in vial	Lyophilized in vial	Liquid in vial	Liquid in PFS[a]	Liquid in PFS
Dose Strength[b]	100 mg/1.0 mL per vial	100 mg/1.0 mL per vial	100 mg/1.0 mL per vial; 50 mg/0.5 mL per vial	100 mg/1.0 mL per PFS; 50 mg/0.5 mL per PFS	100 mg/1.0 mL per PFS; 50 mg/0.5 mL per PFS
Composition[c]					
Buffer	10 mM Na phosphate	10 mM Na phosphate	5.6 mM Histidine	5.6 mM Histidine	5.6 mM Histidine
Stabilizer/Tonicifier	8.5% (w/v) Sucrose	8.5% (w/v) Sucrose	4.1 % (w/v) Sorbitol	4.1 % (w/v) Sorbitol	4.1 % (w/v) Sorbitol
Surfactant	0.001% (w/v) PS 80	0.01% (w/v) PS 80	0.015% (w/v) PS 80	0.015% (w/v) PS 80	0.015% (w/v) PS 80
pH	6.0	6.0	5.5	5.5	5.5
Drug Concentration	100 mg/mL	100 mg/mL	100 mg/mL	100 mg/mL	100 mg/mL

[a] PFS = Pre-filled syringe
[b] Corresponds to amount withdrawable and does not include overfill.
[c] The target composition of the lyophilized DP corresponds to the calculated composition after reconstitution with 1 mL water for injection. For Phase 1 and Phase 2 studies these values correspond to target composition of the diafiltration buffer used during manufacture of Simponi formulated bulk. For Phase 3 product, the concentration of excipients was experimentally determined.

(Continued)

(e)



Simponi (golimumab) 2.3.P.2 Pharmaceutical Development

Table 2: History of Simponi DP development

table 96%

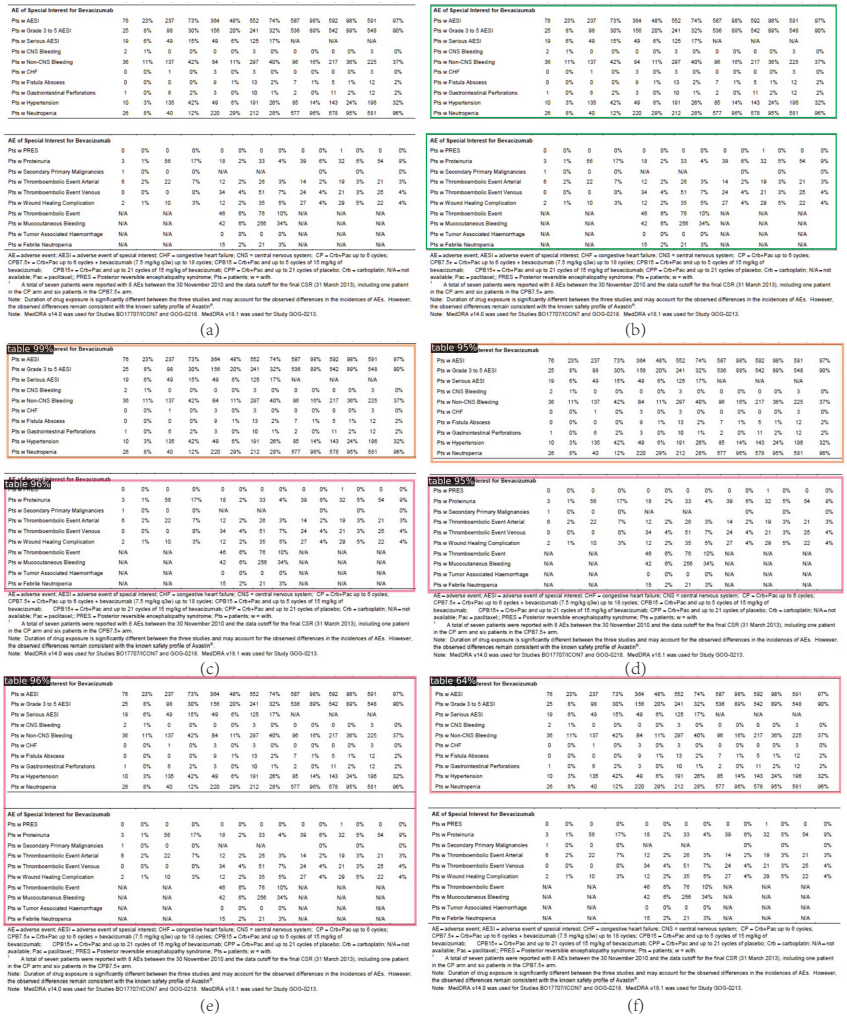
	Phase 1	Phase 2	Phase 3		Commercial
Dosage Form	Lyophilized in vial	Lyophilized in vial	Liquid in vial	Liquid in PFS[a]	Liquid in PFS
Dose Strength[b]	100 mg/1.0 mL per vial	100 mg/1.0 mL per vial	100 mg/1.0 mL per vial; 50 mg/0.5 mL per vial	100 mg/1.0 mL per PFS; 50 mg/0.5 mL per PFS	100 mg/1.0 mL per PFS; 50 mg/0.5 mL per PFS
Composition[c]					
Buffer	10 mM Na phosphate	10 mM Na phosphate	5.6 mM Histidine	5.6 mM Histidine	5.6 mM Histidine
Stabilizer/Tonicifier	8.5% (w/v) Sucrose	8.5% (w/v) Sucrose	4.1 % (w/v) Sorbitol	4.1 % (w/v) Sorbitol	4.1 % (w/v) Sorbitol
Surfactant	0.001% (w/v) PS 80	0.01% (w/v) PS 80	0.015% (w/v) PS 80	0.015% (w/v) PS 80	0.015% (w/v) PS 80
pH	6.0	6.0	5.5	5.5	5.5
Drug Concentration	100 mg/mL	100 mg/mL	100 mg/mL	100 mg/mL	100 mg/mL

[a] PFS = Pre-filled syringe
[b] Corresponds to amount withdrawable and does not include overfill.
[c] The target composition of the lyophilized DP corresponds to the calculated composition after reconstitution with 1 mL water for injection. For Phase 1 and Phase 2 studies these values correspond to target composition of the diafiltration buffer used during manufacture of Simponi formulated bulk. For Phase 3 product, the concentration of excipients was experimentally determined.

(Continued)

(f)

Figure 3.5: Prediction samples of the baseline models on the Open-Tables test set. Figures (a) (b) (c) (d) (e) (f) are the original document image, the ground truth boxes, and the results of TableDet, DiffusionDet, Deformable-DETR, and Sparse R-CNN, respectively. The confidence scores in sub-figures are 100%, 94%, 97% and 96%, respectively.



(a) (b) (c) (d) (e) (f)

Figure 3.6: Prediction samples of the baseline models on the Open-Tables test set. Figures (a) (b) (c) (d) (e) (f) are the original document image, the ground truth boxes, and the results of TableDet, DiffusionDet, Deformable-DETR, and Sparse R-CNN, respectively. The confidence scores in sub-figures are 99%, 96%, 95%, 95%, 96% and 64%, respectively. Notably, there are two prediction boxes in Figures (c) and (d) and one prediction box in Figures (e) and (f).

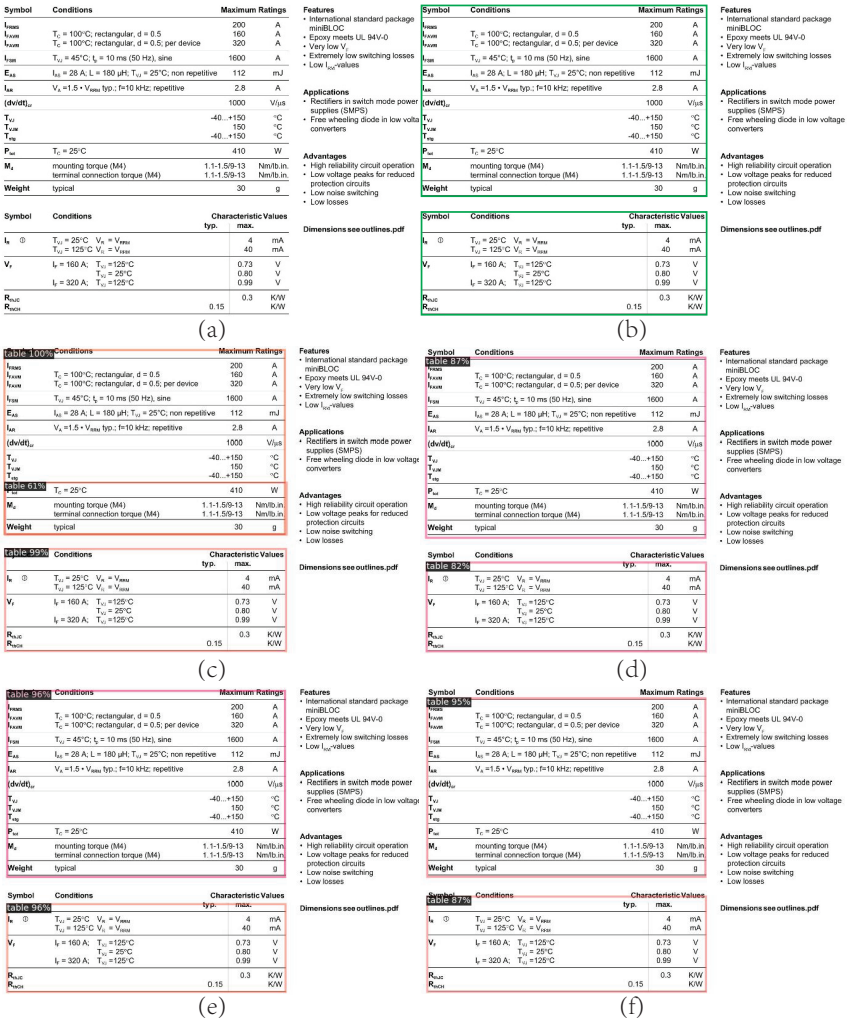


Figure 3.7: Prediction samples of the baseline models on the ICT-TD test set. Figures (a) (b) (c) (d) (e) (f) are the original document image, the ground truth boxes, and the results of TableDet, DiffusionDet, Deformable-DETR, and Sparse R-CNN, respectively. The confidence scores in sub-figures are 100%, 61%, 99%, 87%, 82%, 96%, 96%, 95% and 87%, respectively. Notably, there are three prediction boxes in Figure (c) and two prediction boxes in Figures (d), (e) and (f).

Figure 3.8: Prediction samples of the baseline models on the ICT-TD test set. Figures (a) (b) (c) (d) are the results of TableDet, DiffusionDet, Deformable-DETR, and Sparse R-CNN, respectively. The confidence scores in sub-figures are 100%, 100%, 96%, 94%, 93%, 97%, 73%, and 96%, respectively. Notably, each sub-figure contains two prediction results, in which the upper ones are falsely detected figures.

Table 3.5: Experimental results on the Open-Tables dataset with F1-score. 8834 and 2295 are the number of samples. * means the models are trained with noisy samples.

Dataset		Model	F1 under IoU thresholds				WAvg. F1
Training	Testing		80%	85%	90%	95%	
Open-Tables Training Set (8834)	Open-Tables Test set (2,295)	TableDet	95.8	94.7	91.7	83.2	91.1
		DiffusionDet	97.8	96.6	93.6	84.9	93.0
		Deformable-DETR	96.0	94.8	93.1	87.5	92.6
		Sparse R-CNN	97.3	96.0	93.8	87.9	93.5
Noisy Open-Tables Training Set (8834)	Open-Tables Test set (2,295)	TableDet*	95.4	94.2	91.7	81.7	90.4
		DiffusionDet*	97.1	96.1	93.6	84.8	92.6
		Deformable-DETR*	95.7	94.6	91.9	85.3	91.6
		Sparse R-CNN*	96.8	95.7	93.5	86.6	92.9

3.4.2 Cross Domain Table Detection

This section discusses the potential of using the proposal dataset in a cross-domain setting. As discussed in Section 3.3, ICDAR2013, ICDAR2017, ICDAR2019, Marmot, and TNCR are also manually annotated datasets with high-quality annotations, and their data sources are academic publications and open governmental documents. Therefore, these datasets are merged to form the Open-Tables dataset, which contains 8834 training samples, 1015 validation samples and 2295 testing samples. It is worth mentioning that TNCR has five different groups of tables, as discussed in Section 2.2. All these groups are merged into a single group as tables. After the cleaning tasks discussed in Section 3.2, two cross-domain settings are used to build the benchmarks. First, the ICT-TD's training set is used to train the detection baseline models, and the Open-Tables' test set is used to evaluate the model's performance. The experimental results of this setting are shown in Table 3.6. In the second setting, the training set of the Open-Tables dataset and the test set of the ICT-TD dataset are used to build the benchmarks, and the results are shown in Table 3.7. It is worth mentioning that the same evaluation metrics are used as in section 3.4.1, and the * in Table 3.7 means the models are trained with the noisy version of Open-Tables dataset.

The experimental results show that Deformable-DETR, which performs best for the ICT-TD dataset, also has the best generalization capacity in the cross-domain setting. However, the cross-domain setting is much more challenging, and all the benchmark models' performance degrades by a large margin compared with results in Table 3.4 and Table 3.5.

Table 3.6: Experimental results with F1-score in the cross domain setting. 4000 and 2295 are the number of samples.

Dataset		Model	F1 under IoU thresholds				WAvg. F1
Training	Testing		80%	85%	90%	95%	
ICT-TD Training Set (4,000)	Open-Tables Test set (2,295)	TableDet	80.0	76.6	69.9	51.0	68.7
		DiffusionDet	84.1	80.9	75.1	60.0	74.5
		Deformable-DETR	84.8	82.5	78.2	69.3	78.3
		Sparse R-CNN	82.1	79.2	73.9	62.0	73.8

3.4.3 The Impact of Noise in Open-Tables Dataset

This section presents extra experiments and discusses the impact of noise in the Open-Tables dataset. In the experiments, the training set of Open-Tables with noise is used to train the benchmark models, and the test set of the ICT-TD dataset is used to evaluate the model performance. As shown in Table 3.7, the cleaned version of the Open-Tables dataset can improve the performance of all models, especially when the IoU threshold is above 80%. The experimental results verify the necessity of noise cleaning and label alignment when we create the Open-Tables dataset. Besides, these models with the cleaned Open-Tables test set are also evaluated, and the experimental results are given in Table 3.5. Similar to the results shown in Figure 3.7, the models trained with the cleaned Open-Tables training set perform better than their counterparts trained with the noisy Open-Tables training set. It is worth mentioning that the Open-Tables test set is created by sampling the cleaned test set of ICDAR2013, ICDAR2017, ICDAR2019, and TNCR datasets, as shown in Table 3.2. Therefore, the results shown in Table 3.5 also reflect that these existing datasets can benefit from the proposed Open-Tables dataset.

3.4.4 Comparison with Automatically Generated Datasets

This section presents further experiments to compare the automatically generated datasets with the proposed Open-Tables dataset in the cross-domain setting. More specifically, PubLayNet and TableBank are two large datasets created by parsing meta-data, whereas the Open-Tables is the ensemble of manually labeled small datasets. Since PubLayNet is designed for Layout Analysis, only its annotations on Tables are used in the experiments, resulting in 86,460 training samples. TableBank consists of two sub-sets created by parsing latex and word files, respectively. These two sub-sets are applied separately in the experi-

ments. The experimental results are shown in Table 3.7. Similar to experiments discussed in Section 3.4.3, DiffusionDet, Deformable-DETR, and Sparse R-CNN show better performance than TableDet. The models trained with PubLayNet and TableBank perform worse than the models trained with the Open-Tables, even though PubLayNet and TableBank have a much larger number of training samples than Open-Tables, demonstrating the critical importance of annotation quality in model training. Since DiffusionDet performs best among the models trained PubLayNet and TableBank, some prediction results of DiffusionDet models trained with different datasets are shown in Figure 3.9. These models show different capacities of detecting tables, and the samples in Figure 3.9 show that models trained with automatically generated datasets can more easily fail to detect tables.

3.4.5 Analysis of the Open-Tables Sub-sets

This section discusses the models trained with the Open-Tables dataset and their performance on the Open-Table sub-sets. More specially, the Sparse R-CNN trained with Open-Tables training set is used as an example, and its performance on ICDAR2013, ICDAR2017, ICDAR2019, and TNCR test sets is reported in Table 3.8 because it performances best based on the results in Table 3.5. It is worth mentioning that there are no results on the Marmot dataset because it does not have a test set, as shown in Table 3.2. As shown in Table 3.8, Sparse R-CNN shows promising results on all these sub-sets, especially on the ICDAR2013 dataset. The prediction results on these datasets are further visualized and checked. For Sparse R-CNN, there are two major issues with its prediction results. First, Sparse R-CNN can fail to detect some tables, especially when multiple similar tables are closely distributed, such as the example in Figure 3.6 (f), or the table is relatively small and has a unique style, such as the example in Figure 3.10 (a). Second, Sparse R-CNN can predict inaccurate boxes when the table has no explicit lines, as shown in Figure 3.10 (b). Notably, different models have shown different capacities in detecting tables and can have other issues, as shown in Figures 3.5 and 3.8.

3.4.6 Potential Applications

This section discusses the potential applications of the proposed two datasets. As mentioned earlier, the ICT-TD dataset is created using real documents from the ICT domain. The cross-domain setting that uses Open-Tables' training set to train the models and then test them on the ICT-TD test set does not perform well, as shown in Table 3.7. By contrast, as shown in Table 3.4, the models trained with the ICT-TD training set can achieve

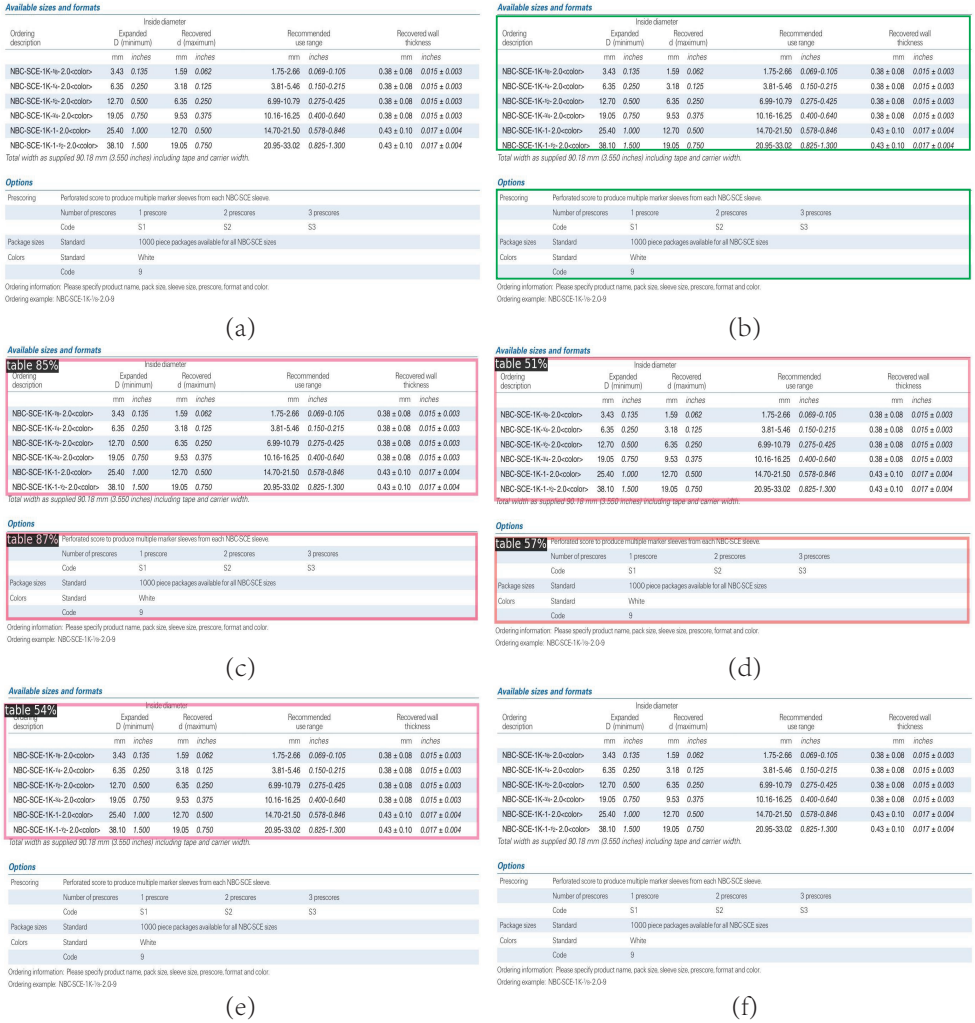
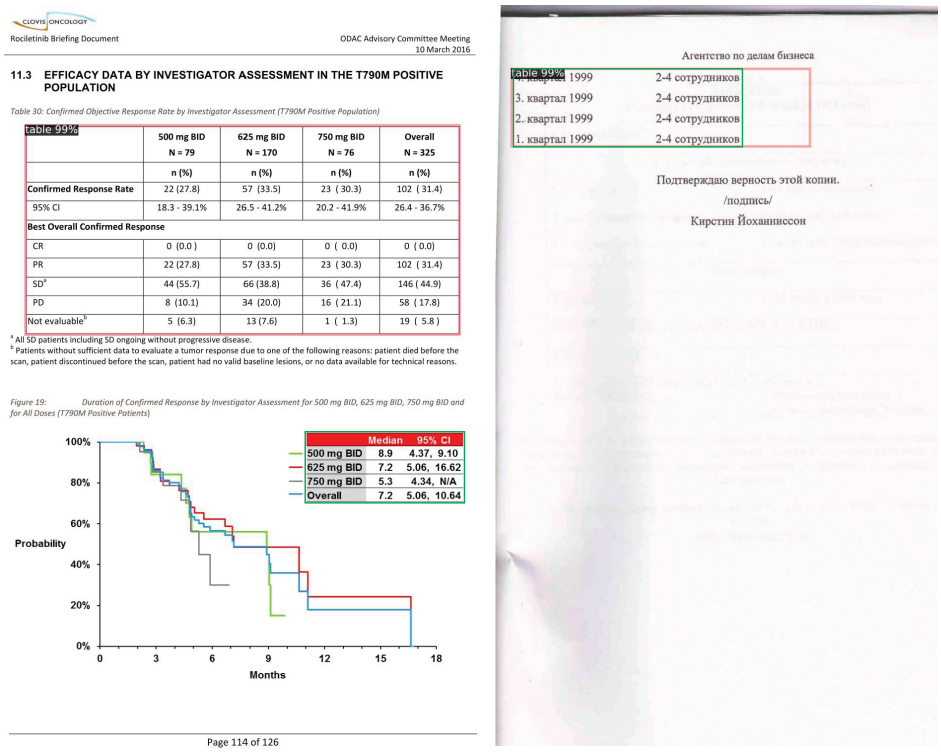


Figure 3.9: Prediction samples of DiffusionDet models. Figures (a) and (b) are the original image and its ground truth. Figures (c), (d), (e) and (f) are the predictions of models trained with the Open-Tables, PubLayNet, TableBank Latex subset, and TableBank Word subset, respectively.



(a) (b)

Figure 3.10: Two failure cases of Sparse-RCNN trained with the Open-Tables training set. These two images are from the TNCR test set. Notably, the green boxes are the ground truth boxes of two failed predictions.

much better results when tested on the ICT-TD test set. Therefore, the proposed ICT-TD dataset can be used to train and evaluate ICT domain-specific models and applied to the ICT supply chain optimization problems [154] as part of the information processing step. Besides, the proposed ICT-TD dataset can also enrich the data sources of public datasets and be used to evaluate models' generalization ability in cross-domain settings. On the other hand, the Open-Tables dataset focuses on addressing the noise issues of existing public datasets. As shown in Table 3.7, a cleaned version of Open-Tables' training set improves the models' generalization ability in the cross-domain settings. Furthermore, the Open-Tables test set is created by merging the cleaned test sets of ICDAR2013, ICDAR2017, ICDAR2019, and TNCR datasets, which means that it can provide more reliable evaluation results.

3.5 Summary of the Chapter

This chapter revisits some popular datasets with high-quality annotations but different annotation definitions, cleans the noisy samples, and aligns the annotations of these datasets to form a larger, high-quality dataset termed Open-Tables. Since the data sources of popular datasets are very limited, a new ICT-TD dataset is proposed using the datasheets from the ICT domain. The proposed ICT-TD dataset contains many domain-specific samples that hardly appear in other open-source datasets, which makes it useful in cross-domain settings. The revisited Open-Tables dataset is consistent and larger, making it more reliable for evaluating the model performance. These two datasets can be more reliable benchmarks to build reliable TD applications that should avoid losing any information in the tables and alleviate the side effects of noisy samples to the model evaluation. At last, strong baselines using state-of-the-art object detection models are built for the ICT-TD dataset and a cross-domain setting. The experimental results show that cross-domain settings are more challenging for the TD problem.

Most existing studies for the TD problem use object detection evaluation metrics that need an IoU threshold. However, these evaluation metrics are indirect to the actual performance of extracting information from tables. For instance, a larger prediction box that can cover all the information of the target table but has a lower IoU score is preferable to the box with a higher IoU score but can lose some information from the target table. Therefore, evaluating models with other metrics can be a good direction for further work to compensate for the drawback of using IoU score based metrics. Besides, the experimental results show that larger datasets do not necessarily result in better performance. Developing automated methods to create, filter, and select proper data for model training

can also be a future direction.

Table 3.7: Experimental results with F1-score in the cross domain setting. 8834, 1000, 86460, 187199 and 73383 are the number of image samples. * means the models are trained with noisy samples.

Dataset		Model	F1 under IoU thresholds				WAvg. F1
Training	Testing		80%	85%	90%	95%	
Open-Tables Training Set (8,834)	ICT-TD Test set (1,000)	TableDet	83.1	80.1	76.1	65.0	75.7
		DiffusionDet	87.9	86.1	81.6	67.0	80.2
		Deformable-DETR	84.1	82.2	79.7	70.1	78.7
		Sparse R-CNN	84.2	81.9	78.5	67.8	77.7
Noisy Open-Tables Training Set (8,834)	ICT-TD Test set (1,000)	TableDet*	81.0	78.2	73.8	61.1	73.1
		DiffusionDet*	86.2	84.0	79.8	66.3	78.6
		Deformable-DETR*	81.9	80.4	77.9	67.7	76.7
		Sparse R-CNN*	85.0	82.5	78.4	64.2	77.1
PubLayNet Training Set (86,460)	ICT-TD Test set (1,000)	TableDet	53.9	50.9	45.6	35.9	46.2
		DiffusionDet	71.8	69.2	62.6	50.8	63.1
		Deformable-DETR	69.2	65.0	58.9	48.6	59.9
		Sparse R-CNN	70.0	66.7	61.2	49.7	61.4
TableBank_{latex} Training Set (187,199)	ICT-TD Test set (1,000)	TableDet	69.9	66.1	61.6	48.1	60.9
		DiffusionDet	79.9	76.9	71.1	56.7	70.6
		Deformable-DETR	78.8	76.2	71.2	57.0	70.3
		Sparse R-CNN	79.0	76.1	70.7	55.4	69.8
TableBank_{word} Training Set (73,383)	ICT-TD Test set (1,000)	TableDet	70.0	68.5	65.3	56.7	64.8
		DiffusionDet	76.4	75.1	71.7	60.1	70.4
		Deformable-DETR	76.0	73.6	70.4	58.6	69.2
		Sparse R-CNN	75.7	73.5	70.5	60.6	69.7

Table 3.8: Experimental results on the sub-sets of Open-Tables. The values are the performance of Sparse R-CNN trained with the Open-Tables training set.

Dataset		F1 under IoU thresholds				WAvg. F1
Training	Testing	80%	85%	90%	95%	
Open-Tables Training Set (8834)	TNCR (1000)	97.6	96.0	93.3	87.5	93.3
	ICDAR2019 (240)	97.5	97.0	95.0	88.7	94.3
	ICDAR2017 (817)	94.9	94.0	92.5	85.9	91.6
	ICDAR2013 (238)	99.3	99.3	98.9	95.0	98.0

Chapter 4

Table Detection

4.1 Motivation

As discussed in Chapter 1, TD is the first step of the proposed pipeline in this thesis, aiming at locating and extracting tables from visually rich document images. Existing state-of-the-art methods [8, 9] usually employ two-stage object detectors with data augmentation and multi-stage transferring learning techniques. However, tables from visually rich documents are usually well-formatted and sparsely distributed, which means that a two-stage detector with dense region proposals might not be necessary. Therefore, this thesis uses Sparse R-CNN [10] as the base model, which leverages sparse learnable region proposals. To improve the base Sparse R-CNN model and avoid information loss for a successful TD model, the following methods are applied to the Sparse R-CNN. First, image size region proposals are applied to cover all the information of target tables in the proposal boxes, and a noise augmentation method is proposed to enrich the diversity of proposal boxes. Second, the IoU score is decoupled into two terms: a ground truth coverage term and a prediction coverage term, which can replace the IoU score in the IoU-based loss functions and evaluation metrics. Based on the decoupled IoU, an Information Coverage Score (ICS) loss is defined, which can guide the model to minimize information loss. At last, a SimOTA [15] based many-to-one label assignment approach is leveraged, which can further improve the SimOTA by adopting a dynamic scheduling scheme to adjust the number of positive assignments dynamically. It is worth mentioning that since the Dynamic Heads in the Sparse R-CNN model also contain Multi-head Attention layers, we consider Sparse R-CNN a transformer-based approach.

To sum up, this chapter covers the following aspects of the proposed Sparse R-CNN

based TD method:

1. This chapter introduces a decoupled IoU score, termed Information Coverage Score (ICS), which can reflect the information loss of the prediction boxes when it is used as evaluation metrics and encourage the model to alleviate the information loss when it is used as loss functions.

2. This chapter introduces an improved SimOTA method by adapting a dynamic scheduling scheme and integrating the ICS loss, proposes a Gaussian Noise Augmented Image Size region proposal method, and applies them to the SpareR-CNN model to further improve the performance of the proposed method.

3. Extensive experiments are conducted using IoU-based evaluation metrics and loss functions on various manually annotated datasets to demonstrate the efficiency and effectiveness of our proposed detection model. Then, further experiments are conducted to demonstrate the benefits of the proposed ICS score when it is applied to the TD problem. The experimental results show that the proposed method can consistently outperform the state-of-the-art benchmark models under different evaluation metrics.

The rest of this chapter is organized as follows: Section 4.2 introduces the formal problem definition and the proposed SparseTableDet model. Section 4.3 presents the experiments and discusses the design aspects of the proposed method. At last, the contents of this chapter and future research directions are summarized in section 4.4.

4.2 Proposed Method for Table Detection

4.2.1 Overall Architecture

Following the architecture of Sparse R-CNN [10], the proposed method also consists of an Initialization Module, a Feature Pyramid Network, and a series of Dynamic Heads. The Initialization Module is used to initialize the learnable proposal boxes and the learnable proposal features. Feature Pyramid Network (FPN) [155] is the backbone network to generate image features for every Dynamic Head. The Dynamic Heads are used to do the regression and classification tasks. Dynamic Head $t+1$ takes the image features generated by FPN and the outputs of Dynamic Head t, including the Refined Proposal Features and Refined Proposal Boxes, as the input to further refine the predictions of Dynamic

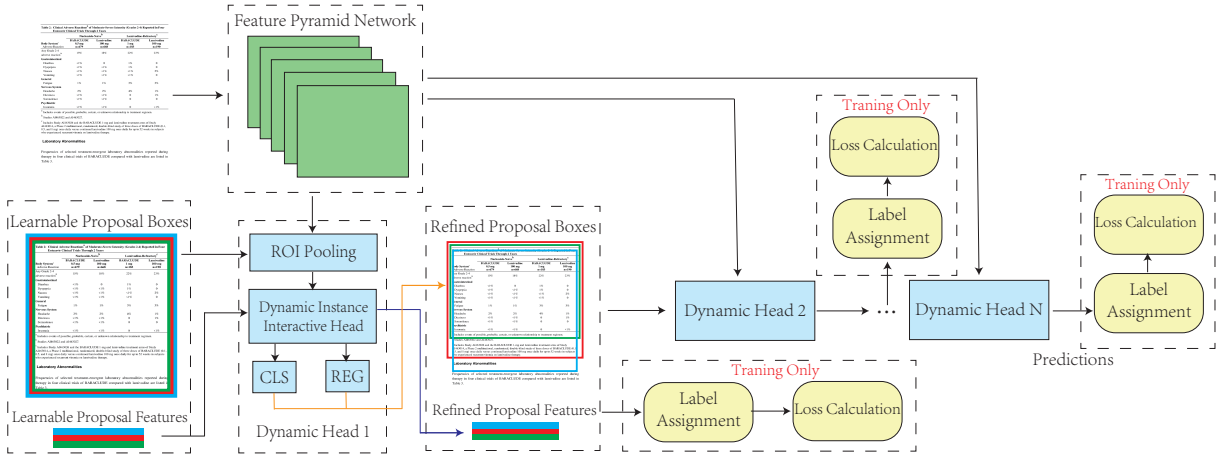


Figure 4.1: The overall architecture of the proposed method. Notably, all Dynamic Heads share an identity structure. For simplicity, only the details of Dynamic Head 1 are shown in this figure.

Head t when $t > 1$. Since the predictions of each Dynamic Head are used to calculate the loss, a label assignment process is operated on these predictions to further calculate the losses. Since the proposed refinements to the Sparse R-CNN model are mainly on the proposal initialization, label assignment, and the loss functions, which will be discussed in section 4.2.2, 4.2.3 and 4.2.4, the default implementations of Sparse R-CNN are kept for other parts which are detailed described in the study [10].

4.2.2 Noise Augmentation to Region Proposals

As discussed in Chapter 1, TD applications typically require predictions to avoid information loss, and the tables in the documents are usually large and have no overlaps. Considering these characteristics of TD applications, using Image Size to initialize the region proposals becomes a good choice compared with other initialization methods, such as Random Initialization [10] and Grid Initialization [10], because it can avoid information loss at the first step of the detector. However, simply using a number of the same proposals may not be optimal. Therefore, this thesis proposes a simple but effective method to augment the region proposals by adding Gaussian Noise to enrich the proposals' diversity. More specifically, assuming that a proposal box is represented by its box center, width, and height, namely $b = \{c_x, c_y, w, h\}$, then the augmented proposal box can be defined as

Equation 4.1, where $\mathcal{N}$ means Gaussian Distribution. In the implementation, boxes are normalized, meaning that an image size box can be represented as $b = \{0.5, 0.5, 1, 1\}$, and μ, σ^2 are set as 0 and 0.01, respectively.

$$b_{aug} = f(\{c_x, c_y, w, h\}) = \{c_x + \epsilon_x, c_y + \epsilon_y, w - 2 \cdot |\epsilon_x|, h - 2 \cdot |\epsilon_y|\}, \epsilon_x \in \mathcal{N}(\mu, \sigma^2), \epsilon_y \in \mathcal{N}(\mu, \sigma^2) \tag{4.1}$$

It is worth mentioning that adding noise to the region proposal boxes can be interpreted as a movement of these boxes. Since initial boxes are set to Image Size, any movement ϵ of the center leads to 2ϵ reduction of height or width, as shown in Figure 4.2. The addition of noise to the Image Size region proposals can enrich the diversity of region proposals, making the model more robust and improving the performance.

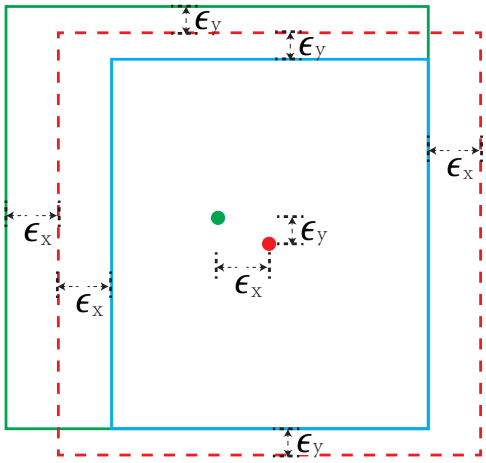


Figure 4.2: A sample of noise augmentation to a region proposal box. The green box is the original box, the dashed red box is the result of center movement, and the blue box is the result box after augmentation.

4.2.3 Many-to-One Label Assignment

As discussed in Chapter 1 and Section 2.3, label assignment plays a key role in the object detection models. Sparse R-CNN employs Hungarian algorithm [26] to perform one-to-one label assignment so that the Non Maximum Suppression (NMS) can be removed from the

processing pipeline. However, as aforementioned, tables in documents are usually large and have no overlaps, meaning that applying NMS to the TD problem does not necessarily lead to drawbacks caused by the NMS, such as the performance degradation caused by the object overlaps [156]. Moreover, the cascaded Dynamic Heads take input proposal features and boxes with different qualities, making it necessary to determine the label assignment dynamically. Many studies [11, 12, 14] have demonstrated that Many-to-One label assignment can bring benefits to the model performance. Therefore, SimOTA [14] is adapted as the base label assignment method.

SimOTA is a simplified version of OTA [11], which can avoid the complex optimization process of OTA. More specifically, SimOTA directly uses the top-k candidates whose centers are in the ground truth bounding boxes as the positive samples, as defined by Equation 4.2, which contains a classification cost, a regression cost, and a center cost. It is worth mentioning that the cost function of Sparse R-CNN is the sum of the cls_cost and regression_cost in Equation 4.2.

$$cost_{SimOTA} = \underbrace{\lambda_{cls} \cdot cost_{cls}}_{cls\_cost} + \underbrace{\lambda_{l1} \cdot cost_{l1} + \lambda_{giou} \cdot cost_{giou}}_{regression\_cost} + \underbrace{\lambda_{center} \cdot cost_{center}}_{center\_cost} \tag{4.2}$$

SimOTA employs a dynamic method to determine the number of positive samples assigned to each ground truth box using the sum of the top 10 IoU scores between a ground truth box and its corresponding prediction boxes without considering the difference of Dynamic Heads. Considering that the inputs of Dynamic Head $t+1$ should contain higher quality boxes than that of t after the refinement of Dynamic Head t, the Dynamic Head $t+1$ should have more positive samples. Therefore, this dynamic method of SimOTA is further extended by adding a scheduling scheme as defined by Equation 4.3, where N is the number of Dynamic Heads, IoU_i is the IoU matrix between the predictions and the ith ground truth, n is a hyperparameter and k_t^i means the number of positive samples assigned to the ith ground truth for Dynamic Head t. Notably, the key consideration for the scheduling scheme is that with the increasing quality of region proposals to the Dynamic Head, the number of assigned proposals to each ground truth can also be larger. Therefore, as defined by Equation 4.3, the quality of region proposals is implicitly estimated by the IoU, and when the t is larger, the number of assigned region proposals also becomes larger.

$$k_t^i = SUM(TOPK(IoU_i, n - 0.5 * (N - t))), t \in [1, N] \tag{4.3}$$

At last, the loss function is defined as the sum of all the Dynamic Heads' loss, as defined

by Equation 4.4. For the classification loss, cross entropy and focal loss [157] are employed for binary and multi-class classification, respectively.

$$\mathcal{L} = \sum_{t=1}^{N} loss_t = \sum_{t=1}^{N} \lambda_{cls} loss_{cls}^t + \lambda_{l1} loss_{l1}^t + \lambda_{giou} loss_{giou}^t \tag{4.4}$$

It is worth mentioning that some studies, such as YOLOX [15] and Dynamic Sparse R-CNN [12], use similar Label Assignment approaches. Dynamic Sparse R-CNN introduces an assignment scheduling scheme to the OTA method [11] to dynamically adjust the number of positive label assignments. However, the OTA method requires a complex optimization procedure, which is significantly more time-consuming than SimOTA. Therefore, SimOTA is leveraged and further improved by adapting the assignment scheduling scheme and the proposed ICS loss function.

4.2.4 Information Coverage Score

As discussed in Chapter 1, the IoU score cannot directly reflect the information coverage of the prediction boxes. This section discusses our proposed decoupled IoU, the Information Coverage Score (ICS). Assume that G and P are the ground truth and prediction boxes, respectively. IoU is the ratio of the intersection of G and P to the union of G and P, as defined by Equation 4.5. In contrast, ICS contains a ground truth coverage term (GT_Coverage) and a prediction coverage term (Pred_Coverage), as defined by Equation 4.6. The ground truth coverage term is the ratio of the intersection of G and P to the G, which can directly measure the information covered by the prediction box. Similarly, the prediction coverage score is defined as the ratio of the intersection of G and P to the P. Since the ICS consists of the GT_Coverage and Pred_Coverage balanced by a hyper-parameter λ, setting a higher λ value can encourage the model to predict boxes covering larger portions of ground truth, which is preferable for the TD task, as discussed in Chapter 1. Figure 4.3 shows three cases for the calculation of ICS, in which green boxes, red boxes, and yellow areas represent the ground truth boxes, prediction boxes, and their intersection areas. It is worth mentioning that the proposed ICS can be used to replace IoU in a variety of IoU-based loss functions, such as GIoU loss [66] and DIoU loss [69]. A simple ICS loss can be defined as Equation 4.7.

$$IoU = \frac{|G \cap P|}{|G \cup P|} \tag{4.5}$$

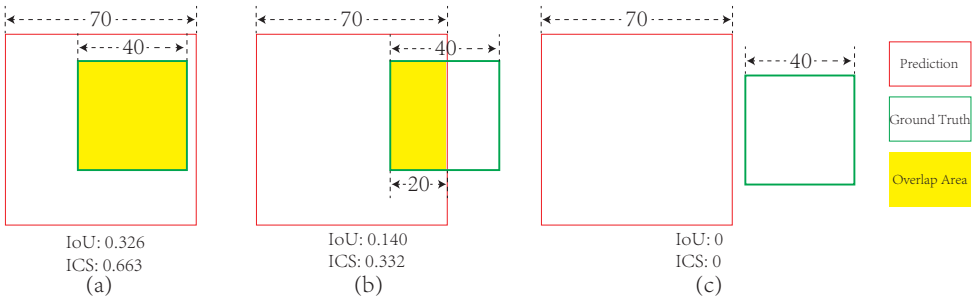


Figure 4.3: Three cases for the Information Coverage Score. λ is set to 0.5. All the boxes are squares.

$$ICS = \underbrace{\lambda \frac{|G \cap P|}{|G|}}_{GT\_Coverage} + \underbrace{(1-\lambda)\frac{|G \cap P|}{|P|}}_{Pred\_Coverage} \tag{4.6}$$

$$ICS\_loss = 1 - ICS \tag{4.7}$$

4.3 Experimental Results and Analysis

This section compares the proposed method with state-of-the-art models using IoU-based evaluation metrics and loss functions. Then, experiments are conducted to demonstrate the benefits of using ICS as the evaluation metrics and loss functions. Lastly, an ablation study is conducted to demonstrate the effectiveness of the proposed many-to-one label assignment method and Gaussian Noise Augmented Image Size region proposal.

4.3.1 Experiment settings and Main results

Many datasets have been proposed for the TD problem. These datasets can be categorized into two groups: human-annotated datasets and generated datasets by parsing meta-data. The former dataset type usually has higher-quality annotations, but the number of samples is usually limited. In contrast, the latter type can have a large number of samples but often contain much noise. In this chapter, only the datasets with high-quality annotations are considered, including ICDAR2017 [2], ICDAR2019 [3], TNCR [7] and ICT-TD datasets.

The TD problem is often the pre-processor step of the information extraction tasks, which requires models to avoid missing tables or predicting other document components as tables. Therefore, the widely used evaluation mAP [28] for the detection models cannot fulfill this requirement. Hence, Precision, Recall, and F1 scores are used as evaluation metrics, which are also widely used in other studies [1–3, 7]. However, the IoU thresholds for these metrics vary among studies, making it hard to compare these models directly. Therefore, in this study, the evaluation metric is aligned for all datasets with Weighted Average F1, as defined in Equation 8, whose thresholds are 60%, 70%, 80% and 90%. The detailed results containing Precision, Recall and F1 under the IoU thresholds from 50% to 95% with a 5% interval are attached in appendix .2.3. Notably, the evaluation metric used here is identical to the one used in ICDAR2019 competition [3] and other evaluation metrics, such as mAP. The metrics in other competitions [1, 2] can be found in the detailed experimental results in appendix .2.2.

$$\text{Weighted Avg. F1} = \frac{\sum_{i=1}^{4} IoU_i \cdot F1@IoU_i}{\sum_{i=1}^{4} IoU_i} \tag{4.8}$$

Table 4.1: Key training parameters of the proposed model.

Parameter	Value	Description
IMS_PER_BATCH	16	number of training samples in an iteration
MAX_ITER	40,600	total number of mini-batch
STEPS	29,000	the mini-batch to apply the learning rate schedule
SCHEDULER	MultiStepLR	the scheduler to change the learning rate
BASE_LR	2.5e-05	the learning rate before applying the scheduler
WEIGHTS	r50_300pro_3x_model	initialization weight of the model
NUM_HEADS	6	the number of Dynamic Head
NUM_PROPOSALS	300	number of region proposals
OPTIMIZER	AdamW	the optimizer to train the model
LABEL_N	8	the hyper parameter N defined by Equation 4.3
NOISE_MEAN	0	mean value of the Gaussian Noise
NOISE_VAR	0.01	variance value of the Gaussian Noise
NMS_THRESH	0.9	non-maximum suppression threshold

For the benchmark models, all the types of popular object detection models discussed in section 2.3 are included, including FasterR-CNN [56], MaskR-CNN [57], TableDet [8], DiffusionDet [13], Deformable-DETR [61], Sparse R-CNN [10], RetinaNet [157], FCOS [55], YOLOX-X [15], YOLOR-X [158], YOLOv5-X [159], YOLOv7-X [160], YOLOv8-X [161].

Table 4.2: Experimental results on ICDAR2017 dataset.

Model	F1 under IoU thresholds				Weighted Avg. F1
	IoU(60%)	IoU(70%)	IoU(80%)	IoU(90%)	
RetinaNet	96.0	93.4	91.7	87.3	91.6
FCOS	96.0	93.8	92.1	87.6	91.9
YOLOX-X	97.7	95.5	92.3	80.0	90.4
YOLOR-X	97.1	95.3	93.4	89.7	93.5
YOLOV5-X	98.5	96.8	95.4	91.9	95.3
YOLOV7-X	97.6	96.3	94.6	91.5	94.6
YOLOV8-X	97.9	96.2	95.3	92.5	95.2
FasterR-CNN	97.1	96.0	93.8	89.6	93.7
MaskR-CNN	96.8	96.0	94.8	91.1	94.4
TableDet	98.8	97.1	95.0	90.4	94.9
DiffusionDet	98.3	97.0	94.9	90.3	94.7
Deformable-DETR	97.5	96.7	94.4	91.4	94.6
Sparse R-CNN	98.3	97.9	96.1	94.0	96.3
SparseTableDet	**99.5**	**99.4**	**98.2**	**94.8**	**97.7**

FasterR-CNN, MaskR-CNN, TableDet, DiffusionDet, Deformable-DETR, and Sparse R-CNN are trained with 120 epochs, and other one-stage detectors are trained with 300 epochs. The detailed settings of these benchmark models are included in Chapter .2.1. The proposed SparseTableDet is built on the code base of Sparse R-CNN, and the key training parameters are summarized in Table 4.1. It is worth mentioning that the parameter names in Table 4.1 are aligned with the names in Detectron2 [150]. More specifically, IMS_PER_BATCH is the total number of training samples in an iteration. MAX_ITER and STEPS refer to the total number of mini-batch used in the training and the mini-batch to apply the learning rate scheduler, respectively. WEIGHTS is the initialization weight of the model. In our implementation, the Sparse R-CNN model pre-trained with COCO dataset [28] is used as the initialization weight. NUM_HEADS and LABEL_N are two custom parameters in the proposed SparseTableDet, which refer to the number of Dynamic Head and the hyperparameter N defined by Equation 4.3, as discussed in section 4.2. At last, the model is trained with AdamW [162] optimizer. For the training of all models, we use the validation set for the model selection and hyperparameter tuning.

The experimental results for the ICDAR2017, ICDAR2019, TNCR and ICT-TD datasets are shown in Table 4.2, 4.3, 4.4 and 4.5, respectively. The experimental results show that

Table 4.3: Experimental results on ICDAR2019 dataset.

Model	F1 under IoU thresholds				Weighted Avg. F1
	IoU(60%)	IoU(70%)	IoU(80%)	IoU(90%)	
RetinaNet	98.0	96.7	94.5	86.8	93.4
FCOS	97.6	96.5	93.6	85.7	92.7
YOLOX-X	97.1	96.0	94.6	89.2	93.8
YOLOR-X	98.6	98.2	97.2	93.4	96.6
YOLOV5-X	99.0	98.9	98.2	95.7	97.8
YOLOV7-X	99.2	98.6	98.0	94.1	97.2
YOLOV8-X	99.2	99.1	98.1	94.7	97.5
FasterR-CNN	97.4	96.2	95.0	90.4	94.4
MaskR-CNN	98.2	97.0	95.8	91.9	95.4
TableDet	98.1	96.8	94.9	91.5	94.9
DiffusionDet	98.9	97.4	95.8	91.3	95.5
Deformable-DETR	98.4	97.9	96.5	92.7	96.0
Sparse R-CNN	98.6	98.1	97.5	94.9	97.1
SparseTableDet	**99.3**	**99.1**	**98.9**	**96.3**	**98.3**

the proposed SparseTableDet can consistently outperform the state-of-the-art benchmark models regarding the Weighted Average F1 score. The proposed model is also compared with other state-of-the-art models optimized for the TD problem following the evaluation protocols of ICDAR2013, ICDAR2017, and ICT-TD datasets, and the results are in appendix .2.2. With these competition evaluation protocols, the proposed method can still consistently outperform the benchmark models.

4.3.2 ICS for model training and evaluation

As discussed in section 4.2.4, GT_Coverage term in the ICS is a direct metric to measure whether the prediction box covers all the target content. This section uses the GT_Coverage as the evaluation metric to evaluate the model performance. More specifically, the IoU score defined in Equation 4.8 is replaced with GT_Coverage to define a new Weighted Average F1 score as the evaluation metric, as defined by Equation 4.9. To demonstrate the effectiveness of ICS as the loss function, the cost_giou in Equation 4.2 and GIoU loss in Equation 4.4 are replaced with their ICS-based counterparts. For simplicity, M_{giou} and M_{ics} are used to represent the model trained with GIoU loss and ICS loss, respectively. As

Table 4.4: Experimental results on TNCR dataset.

Model	F1 under IoU thresholds				Weighted Avg. F1
	IoU(60%)	IoU(70%)	IoU(80%)	IoU(90%)	
RetinaNet	92.7	92.0	90.6	84.8	89.6
FCOS	90.8	89.9	88.8	83.3	87.8
YOLOX-X	90.6	89.3	86.1	79.6	85.8
YOLOR-X	94.2	93.4	91.8	86.4	91.0
YOLOV5-X	95.8	95.5	94.	89.6	93.5
YOLOV7-X	95.2	95.0	93.7	89.3	93.0
YOLOV8-X	96.1	95.5	94.6	90.1	93.7
FasterR-CNN	91.5	91.0	90.3	84.4	88.9
MaskR-CNN	92.5	92.2	90.9	84.7	89.6
TableDet	94.7	94.4	93.3	87.7	92.2
Deformable-DETR	94.4	94.1	92.9	89.3	92.4
DiffusionDet	95.4	94.6	93.1	88.5	92.5
Sparse R-CNN	95.1	94.9	94.4	90.9	93.6
SparseTableDet	**96.3**	**96.2**	**95.8**	**92.7**	**95.1**

shown in Table 4.6, when the IoU-based metrics are used, M_{giou} can perform better than M_{ics}. However, as aforementioned in section 1 and 4.2.4, GT_Coverage term defined in the ICS is a direct measure to evaluate the ground truth information covered by the prediction. GT_Coverage-based evaluation metric is used, M_{ics} can perform better. Figure 4.4 shows two prediction results of M_{giou} and M_{ics}. Using an ICS-based loss function can encourage the model to alleviate the information loss during the optimization process because the GT_Coverage term in the ICS is a direct measure of the information loss and is more sensitive to the information loss. Some other prediction samples of these two models are in appendix .3. Notably, bias might be introduced when using the GT Coverage score as the evaluation metric because the GT Coverage score cannot reflect the difference of prediction boxes once the predictions can cover the ground truth. However, the proposed ICS and GT Coverage scores can provide more insights regarding the quality of predictions and complement IoU-based metrics.

$$\text{Weighted Avg. F1} = \frac{\sum_{i=1}^{4} GT\_Coverage_i \cdot F1@GT\_Coverage_i}{\sum_{i=1}^{4} GT\_Coverage_i} \tag{4.9}$$

Table 4.5: Experimental results on the ICT-TD dataset.

Model	F1 under IoU thresholds				Weighted Avg. F1
	IoU(60%)	IoU(70%)	IoU(80%)	IoU(90%)	
RetinaNet	95.8	93.6	91.0	83.7	90.4
FCOS	91.8	90.4	87.9	82.3	87.6
YOLOX-X	95.8	93.6	90.1	81.6	89.5
YOLOR-X	97.5	96.0	94.3	89.0	93.8
YOLOV5-X	98.0	97.2	95.8	91.7	95.3
YOLOV7-X	98.6	97.6	95.7	92.6	95.8
YOLOV8-X	97.9	97.2	95.6	92.3	95.4
FasterR-CNN	96.8	94.7	92.9	86.8	92.3
MaskR-CNN	96.2	94.8	92.8	87.9	92.5
TableDet	96.9	95.7	93.6	89.1	93.4
DiffusionDet	97.6	96.8	95.5	91.1	94.9
Deformable-DETR	97.4	96.5	95.0	91.2	94.7
Sparse R-CNN	97.1	95.9	94.3	90.4	94.1
SparseTableDet	**98.2**	**97.9**	**97.2**	**94.2**	**96.7**

4.3.3 Ablation Study

This section discusses the effectiveness of the proposed Image Size region proposals, Noise Augmented Proposals, and the Many-to-One label assignment method. Sparse R-CNN acts as the baseline model in this section, which uses Hungarian Matching for the label assignment and random region proposals. The ICDAR2019 dataset is used to conduct experiments and use the Weighted Average F1 scores defined by Equation 4.8 and Equation 4.9 as the evaluation metrics. It is worth mentioning that 60%, 70%, 80% and 90% are chosen as thresholds to align with the metric in section 4.3.1. The experimental results are shown in Table 4.7 and 4.8, where Sparse R-CNN(R), Sparse R-CNN(I) are the Sparse R-CNN initialized with the random proposals and image size proposals, respectively. ManytoOne and Noise represent the proposed many-to-one label assignment and the Noise Augmentation to region proposals. The experimental results show that using image size region proposals, adding noise to the region proposals, and Many-to-One label assignment can improve the performance of the base Sparse R-CNN model.

(a) ICS-based loss (b) IoU-based loss

Figure 4.4: Prediction samples of models trained with ICS-based loss and IoU-based loss.

4.4 Summary of the Chapter

In this chapter, Sparse R-CNN [10] based method is proposed, which is further improved by introducing Noise Augmented region proposal generation, Many-to-One label assignment, and a decoupled IoU. The experimental results show that the proposed method can consistently outperform benchmark models regarding the Weighted Average F1 score on various datasets. Furthermore, considering the requirement of TD applications, GT_Coverage in ICS is used to replace IoU to act as the evaluation metric and ICS is used to replace IoU to derive ICS-based loss functions. The experimental results demonstrate that the GT_Coverage can be a better metric reflecting the prediction's information loss, and ICS-based loss can guide models to cover more information of the target objects. In this chapter, all the area in a ground truth box is assumed to contain information without considering the inner structure of tables. However, some tables contain extra spaces, meaning that some smaller prediction boxes than their ground truth boxes do not lead to any information loss. Therefore, it can be a direction to consider the inner structure of a table to build more reliable evaluation metrics for the TD applications. Besides, as aforementioned in section 4.3.2, the proposed GT_Coverage score cannot reflect the difference in box size once the prediction box can cover the whole ground truth. Therefore, it can be another direction to integrate the size of boxes to the GT_Coverage score to make it more versatile. Moreover, as mentioned in Section 3.4.5, Sparse R-CNN may fail for some cases, such as

Table 4.6: Experimental results on ICDAR2019 dataset evaluated by Weighted Average F1 scores using GT_Coverage and IoU as thresholds.

Loss Function	Metric	F1 under IoU thresholds				Weighted Avg. F1
		60%	70%	80%	90%	
GIoU	IoU	99.3	99.1	98.9	96.3	98.3
ICS	IoU	99.4	98.8	98.1	92.9	97.0
GIoU	GT_C	99.6	99.5	99.1	98.3	99.1
ICS	GT_C	99.7	99.6	99.5	98.6	99.3

Table 4.7: The effectiveness of each component using IoU scores as thresholds.

	F1 thresholded by IoU				Weighted Avg. F1
	60%	70%	80%	90%	
Sparse R-CNN (R)	98.6	98.1	97.5	94.9	97.1
Sparse R-CNN (I)	99.1	98.7	98.2	95.3	97.6
I+ManytoOne	99.4	99.1	98.8	95.3	97.9
I+Noise+ManytoOne	99.3	99.1	98.9	96.3	98.3

the case containing multiple tables closely distributed. For the proposed method in this chapter, the prediction may sometimes generate a large prediction box covering all the tables together. At last, as shown in Section 4.3, even though the proposed method can consistently outperform benchmark models, sometimes the margin is small, making the measurement of uncertainty in these models important. However, since some benchmark models did not provide pre-trained weights to reproduce their results, these models need to be reproduced from scratch. Therefore, this thesis leaves the uncertainty analysis as future work.

Table 4.8: The effectiveness of each component using GT Coverage scores as thresholds.

	F1 thresholded by GT_C				Weighted Avg. F1
	60%	70%	80%	90%	
Sparse R-CNN (R)	98.7	98.7	97.9	96.7	97.9
Sparse R-CNN (I)	99.1	99.0	98.3	97.6	98.4
I+ManytoOne	99.4	99.4	99.2	98.1	98.9
I+Noise+ManytoOne	99.6	99.5	99.1	98.3	99.1

Chapter 5

Table Structure Recognition

5.1 Motivation

As discussed in Chapter 1, Table Structure Recognition (TSR) aims to transform table images into semi-structured or structured formats. Existing TSR models can be roughly categorized into three groups: image-to-sequence, graph-based and detection-based models. Considering the complexities of these types of approaches, this chapter focuses on the detection-based models.

Many studies have discussed the detection-based TSR methods [22–24, 86]. However, these methods have the following issues. First, the problem formulations cannot fully represent the complex structures of tables. For example, some studies [22, 23, 86] do not define Headers as detection targets, making it impossible to transform the header information. Figure 5.1a shows the problem formulation of study [86], which can infer the spanning cells but cannot provide header information. Second, even though some studies define all necessary detection targets, they ignore that their formulations are multi-label detection problems, which must be considered in the detection model design. For example, Figures 5.1b and 5.1c are the definitions of PubTables1M [24], which define Tables, Columns, Rows, Spanning Cells, Projection Row Headers, and Column Headers. However, the projected row headers can share identical bounding boxes with rows, as shown in Figure 5.1b, and the column header can share an identical bounding box with a defined row, as shown in Figure 5.1c. Therefore, this multi-label detection definition needs to be considered in the detection model design. Third, studies employing two-stage object detection models for the TSR task are not specifically designed based on the characteristics of table images. Table images from visually rich documents can have extreme aspect ratios and

dense detection targets. Therefore, it is necessary to properly tune and design a regional proposal network of two-stage detectors. Besides, the difference between using two-stage and transformer-based detectors is also not well-explored. Fourth, detection-based TSR solutions usually apply detection metrics, such as COCO metrics, to evaluate the model performance. However, COCO metrics are not aligned with TSR metrics, such as TEDS. Lastly, the role of feature extraction in the detection models for the TSR task has not been well-explored. Deformable convolution is widely used in detection models to improve detection performance. However, sometimes, it can degrade the TSR performance in terms of the TEDS score.

Therefore, this chapter discusses and explores the underlying reasons for these issues of existing studies and proposes simple methods to alleviate these issues. To sum up, this chapter covers the following aspects:

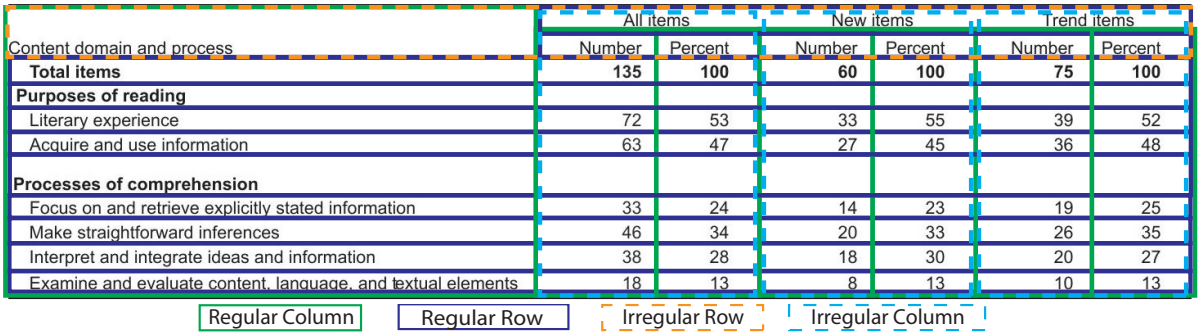
1. This chapter comprehensively revisits existing detection-based TSR models and explores possible reasons hindering the performance of these models, including the improper problem formulation, the mismatch issue of detection metrics and TSR metrics, the inherent characteristics of detection models, and the impact of feature extraction. The analysis and findings can be a guideline for further improving the performance of detection-based TSR models.

2. Based on the analysis and findings, three simple methods are applied to improve Cascade R-CNN, including proposing a pseudo-class generation method to transform multi-label detection into a regular single-label detection problem, adjusting the ratio aspects and the number of regional proposals in the region proposal generation, applying the deformable convolution and introducing a Spatial Attention Module to build the long-range dependencies and context information in the backbone network.

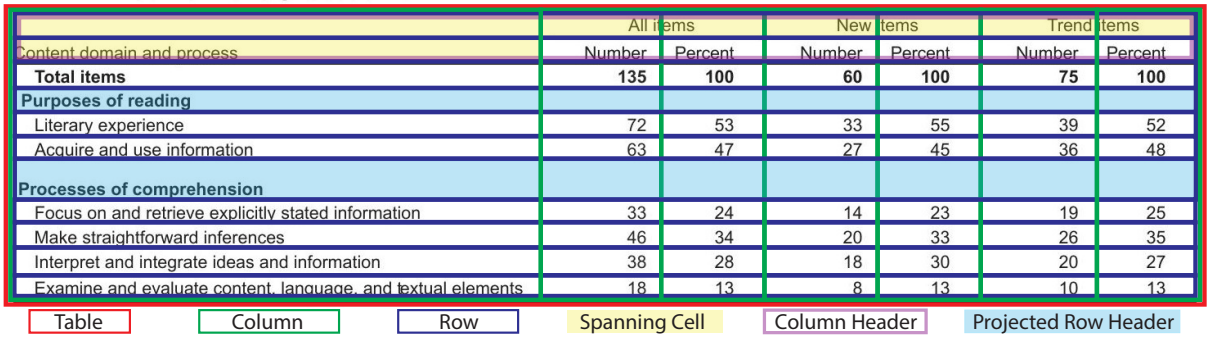
3. Extensive experiments are conducted to evaluate the proposed solution on various datasets, including SciTSR [79], FinTabNet [81], PubTabNet [21] and PubTables1M [24] with both detection metrics and cell-level TSR metrics. The experimental results show that the proposed solution can outperform state-of-the-art models in terms of detection and cell-level TSR metrics.

4. The analysis and findings are further verified with experiments. The insights from the experimental results are discussed and summarized for further model design.

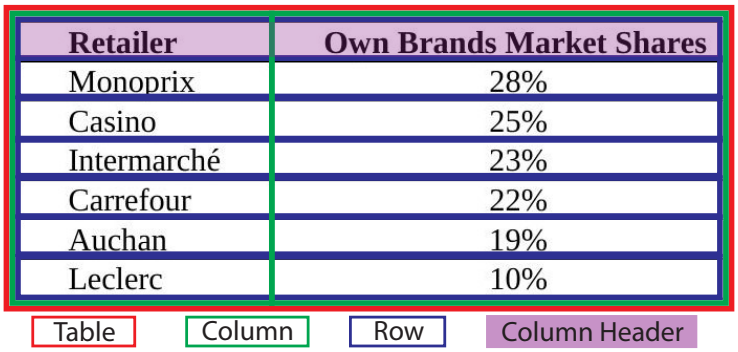
The rest of this chapter is organized as follows: Section 5.2 revisits existing detection-based solutions. Section 5.3 describes our proposed solution. Section 5.4 shows the experimental



(a) Four types of defined table components in [86]. This definition can infer spanning cells but cannot provide header information. Besides, the projected row headers are treated as rows, which can lead to a wrong structure.



(b) Six types of defined table components in [24]. The defined projected row headers in this sample share identical bounding boxes with two corresponding rows.



(c) Four types of defined table components in [24]. In this sample, the defined column header shares an identical bounding box with a defined row.

Figure 5.1: Different problem formulations for the detection-base TSR.

results and discusses the design aspects of the proposed method. At last, Section 5.6 includes the summary of this chapter and possible research directions.

5.2 Rethinking Detection-based TSR Models

5.2.1 Preliminaries

Since most existing detection-based TSR models are based on two-stage and transformer-based detectors, Cascade R-CNN [25] and Sparse R-CNN [10] are used as two examples of these two types of detectors. This section briefly reviews their critical designs.

Cascade R-CNN

Cascade R-CNN [25] is a typical two-stage detection model containing a Backbone Network, a Region Proposal Network (RPN), and a series of Cascade Heads, as shown in Figure 5.2. The RPN is the first regression step of a two-stage detection model responsible for generating region proposals. More specifically, a set of predefined anchor boxes are defined and slides across the feature map to generate the fix-length of feature vectors for the classification and regression tasks in the RPN [56]. The classification task classifies anchor boxes into object and background, and the regression task coarsely regresses the anchor boxes to generate higher-quality region proposals. Since the RPN only coarsely classifies and regresses the anchor boxes, the parameters of defining anchor boxes play a key role in the performance of the RPN, such as the number of anchor boxes, the aspect ratios of anchor boxes, and the scales of applied feature maps.

The Backbone Network is used to extract features of the input images, which is often followed by Feature Pyramid Network (FPN) [155] to extract and fuse features from different scales. The extracted features, together with the region proposals generated by the RPN, are fed into the first cascade head for the classification and regression tasks, and the regression results would be the inputs of the subsequent Cascade Head, as shown in Figure 5.2. Since there are multiple Cascade Heads, all the outputs of these Cascade Heads are used to calculate the loss. Moreover, the final loss of the model can be defined as the sum of these Cascade Heads loss and the RPN loss, as defined by Equation 5.1, where N is the number of Cascade Heads. It is worth mentioning that there are three Cascade Heads in Figure 5.2 following the most popular Cascade R-CNN model. Each Cascade Head has a REG Head and a CLS Head for the regression and classification tasks, respectively. The

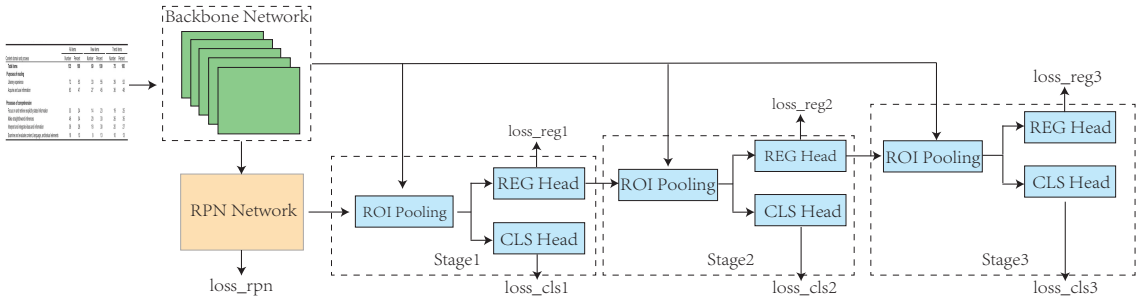


Figure 5.2: Overall architecture of Cascade R-CNN.

input features of these REG Heads and CLS Heads e_{cls}, e_{reg} are extracted by applying ROI Pooling operations to the features from Backbone Network with the proposal boxes b, which can be defined by Equations 5.2 and 5.3 where $PROJ, ROI\_POOL$, and $BACKBONE$ are the Projection layer, ROI Pooling operations, and the Backbone Network. Therefore, for a trained model, the input features of the CLS Heads e_{cls} are determined by the input image x and the proposal boxes b, meaning that a single proposal box cannot be classified into multiple classes because CLS Heads are not multi-label classifiers.

$$\mathcal{L} = \mathcal{L}_{rpn} + \sum_{i=1}^{N} (\mathcal{L}_{cls}^{i} + \mathcal{L}_{reg}^{i}) \tag{5.1}$$

$$e_{cls} = PROJ_{cls}(ROI\_POOL(BACKBONE(x), b)) \tag{5.2}$$

$$e_{reg} = PROJ_{reg}(ROI\_POOL(BACKBONE(x), b)) \tag{5.3}$$

Sparse R-CNN

Sparse R-CNN is a popular end-to-end transformer-based detection model. Similar to Cascade R-CNN, Sparse R-CNN also employs a cascade architecture containing a series of Dynamic Heads, as shown in Figure 5.3. In each Dynamic Head, an ROI Pooling layer is applied to extract features from the feature map based on the given proposal boxes, and the extracted features, together with the learnable proposal features, are fed to the Dynamic Instance Interactive Head to generate final features for the classification and regression

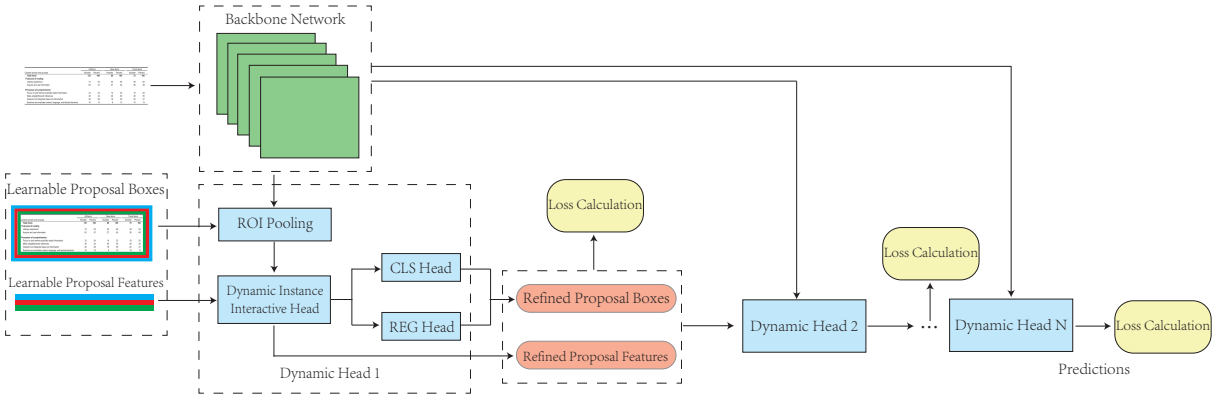


Figure 5.3: Overall architecture of Sparse R-CNN.

tasks. Therefore, the features fed into CLS Head and REG Head of each Dynamic Head can be defined as Equations 5.4 and 5.5, where $BACKBONE$, DYN_HEAD, and $PROJ$ are the Backbone Network, Dynamic Instance Interactive Head and the Projection layer, respectively, and x, b, f are the input image, the proposal boxes and the learnable proposal features. It is worth mentioning that Sparse R-CNN does not use any RPN network to generate regional proposals. Instead, it proposes to use a set of learnable proposal boxes paired with a set of learnable features, in which learnable proposal boxes can be initialized by some pre-defined methods, such as image size initialization, random initialization, and grid initialization. Once the model is trained, the proposal boxes can be treated as an identical value, such as the box of image size, and their classification and regression results are mainly determined by their corresponding learnable proposal features f and the input image x. Therefore, for a multi-label detection problem, when objects belonging to different classes can share an identical box, the learnable proposal features can be different for these objects, making it possible for Sparse R-CNN to deal with multi-label detection tasks.

$$e_{cls} = PROJ_{cls}(DYN\_HEAD(ROI\_POOL(BACKBONE(x), b), f)) \tag{5.4}$$

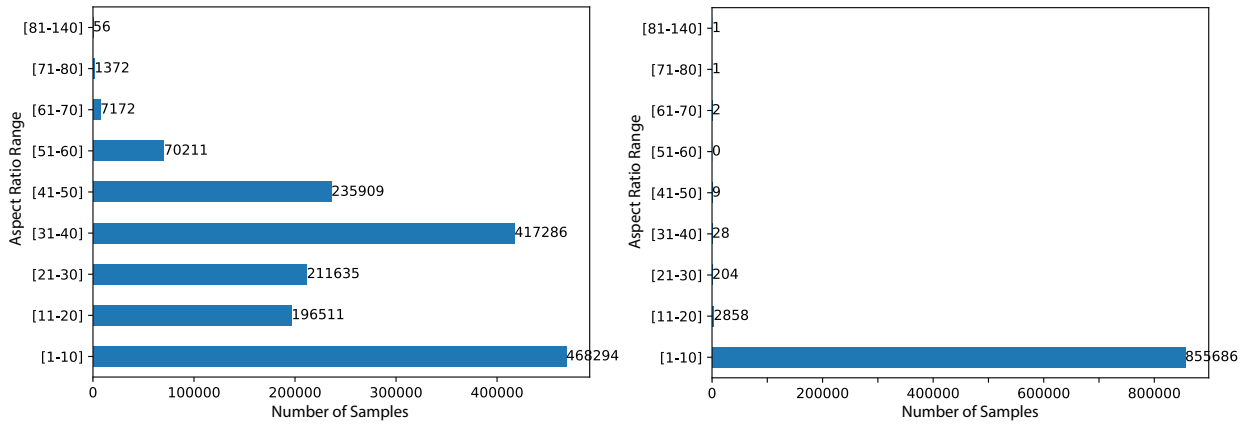
$$e_{reg} = PROJ_{reg}(DYN\_HEAD(ROI\_POOL(BACKBONE(x), b), f)) \tag{5.5}$$

5.2.2 Rethinking Problem Formulations

As aforementioned in Chapter 1 and Section 5.1, there have been many detection-based solutions [22–24,86] with different problem formulations that either oversimplified the TSR task or ignored its multi-label characteristic. More specifically, following image-to-sequence TSR models, a detection-based TSR model should be able to fully reconstruct the structure of both regular and spanning table cells, as well as provide information regarding header cells. However, studies [22, 23] formulate the problem as only detecting columns and rows, making it impossible to deal with spanning cells and identify header cells. TableStrRec [86] further extend the formulation by defining regular column, regular row, irregular column, and irregular row so that the spanning cell can be inferred from these four types of components, as shown in Figure 5.1a. However, this formulation still cannot provide information regarding header cells, which oversimplifies the TSR task. Besides, these formulations treat the Projected Row Header as a regular row, resulting in over-simplified table structures. By contrast, PubTables1M [24] defines six types of components, including Table, Column, Row, Spanning Cell, Column Header, and Projected Row Header, as shown in Figures 5.1b and 5.1c, which can provide as much structure information as image-to-sequence TSR models. However, this formulation does not consider that some Column Headers and Projected Row Headers can share identical bounding boxes with corresponding Rows. For example, as shown in Figure 5.1b, the bounding boxes of the two Projected Row Headers can also be classified as Rows. Similarly, as shown in Figure 5.1c, the Column Header's bounding box is also the Row's bounding box. Therefore, the problem definition of study [24] is a multi-label detection problem, which must be considered when we choose and design detection models. It is worth mentioning that all these problem formulations use extracted table images as inputs. Even though many studies [163] have achieved very promising performance on the Table Detection (TD) task, it is still difficult to guarantee that all the table content can be fully included in the detection results. Therefore, in practice, the detected bounding boxes of tables from the TD model are often padded with extra pixels, making it necessary to define a Table component for TSR.

5.2.3 Revisiting Region Proposal Generation

As aforementioned in Section 5.2.1, the parameters of generating anchor boxes in the RPN play a key role in two-stage detection models, while transformer-based detection models, such as DETR and Sparse R-CNN, use learnable queries or proposals without the need to tune the RPN. If two-stage detection models are chosen as based models, such as Cascade R-CNN which is used in TableStrRec [86], it is necessary to identify the difference



(a) Aspect ratios of the FinTabNet training set. (b) Aspect ratios of the COCO training set.

Figure 5.4: Statistics of aspect ratio values of COCO and FinTabNet training sets. When an aspect ratio is less than 1, its multiplicative inverse counts the number of aspect ratios.

between the TSR detection problem and widely discussed common object detection problem, because the default settings of detection frameworks, such as Detectron2 [150] and MMDetection [164] are often tuned on COCO [28] dataset. Therefore, the statistics of the COCO dataset with a popular TSR dataset, FinTabNet [81], are compared regarding the number of objects in each image and the aspect ratios of objects. More specifically, the COCO training set contains 118287 images and 860001 target objects, resulting in an average of 7.27 objects in each image, while the FinTabNet training set contains 78537 images, 1628298 target objects, resulting in an average of 20.73 objects in each image. Besides, the aspect ratios of objects in these two datasets are also very different, as shown in Figure 5.4. The vast majority of target objects in the COCO training set have aspect ratios between 1 and 10, while objects in the FinTabNet training set have much larger aspect ratios. Therefore, it is necessary to consider these differences when tuning the parameters of RPN if a two-stage object detection model is employed for the TSR task, such as increasing the number of region proposals and adjusting the aspect ratios of anchor boxes. On the other hand, transformer-based detection models, such as Sparse R-CNN and DETR, can alleviate the issues caused by these differences intrinsically because they use learnable queries (learnable proposals) instead of an RPN, as discussed in Section 5.2.1. However, increasing the number of learnable queries for each image might also be useful for transformer-based detection models because TSR datasets contain more objects than common object detection datasets.

5.2.4 Rethinking Detection and TSR Metrics

As mentioned in Chapters 1 and 2, detection-based TSR models need a deterministic rule-based post-processing method to transform the detected table components into structured sequences. Existing studies [22,23,86] usually use the detection performance to evaluate the model performance before applying the post-processing method. However, the detection metrics are not aligned with cell-level TSR metrics. COCO [28] and TEDS [21] metrics can be two examples for further analysis. The COCO metrics employ mean Average Precision (mAP) to evaluate the model performance, which can be defined by Equation 5.6 where N, $precision_i(r)$ and dr in Equation 5.6 are the number of classes, and the precision at a given recall level r for class i. In practice, the precision-recall curves in COCO metrics are computed for each class at a series of IoU thresholds, and the integral of $precision_i(r)$ often is approximated by the discrete sum. The IoU score can be defined by Equation 5.7, where $A \cap B$ and $A \cup B$ are the intersection and union of bounding boxes A and B. In practice, in many studies, mAP is represented by AP and calculated by averaging the mean precision scores of all categories at IoU thresholds from 0.5 to 0.95 with 0.05 intervals. AP50 and AP75 are the mean precision scores of all categories at IoU thresholds of 0.5 and 0.75, respectively. Therefore, COCO metrics are IoU-based evaluation metrics. By contrast, TEDS can be defined by Equation 5.8, where EditDist is the tree-edit distance, and T is the number of nodes in the tree, meaning that TEDS is not correlated with IoU scores.

$$mAP = \frac{1}{N} \sum_{i=1}^{N} \left(\int_0^1 \text{precision}_i(r)\, dr \right) \tag{5.6}$$

$$IoU = \frac{A \cap B}{A \cup B} \tag{5.7}$$

$$TEDS(T_a, T_b) = 1 - \frac{EditDist(T_a, T_b)}{max(|\ T_a\ |, |\ T_b\ |)} \tag{5.8}$$

On the other hand, TSR datasets usually use a canonicalization procedure [24] or annotate the bounding boxes following the lines in tables, which makes the ground truth boxes larger than the minimum box that can recover the structure of the table. Figure 5.5 shows an example from the FinTabNet dataset, whose ground truth boxes are larger than the minimum bounding boxes for table structure. Considering the four prediction boxes in Figure 5.5, since the prediction 1 is smaller than the minimum box for table structure, and the prediction2 can cover all content of the minimum box for table structure and has a larger

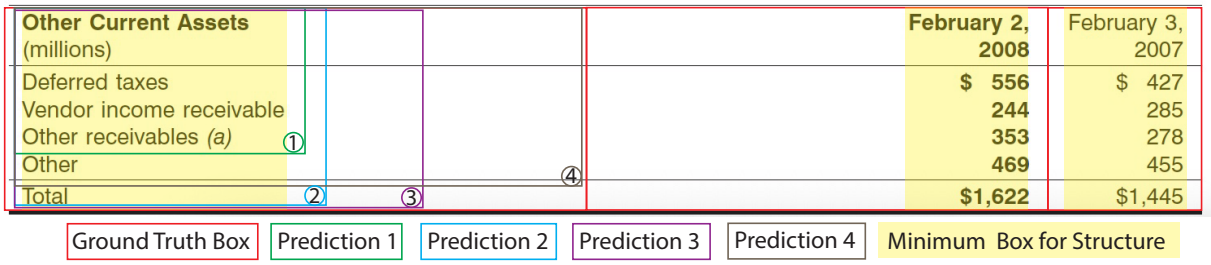


Figure 5.5: A sample from the FinTabNet dataset with ground truth boxes larger than the minimum bounding boxes for table structure. For simplicity, we only show the annotations of columns.

IoU with the ground truth box, prediction 2 can lead to better performance regarding both COCO and TEDS metrics than prediction 1. By contrast, prediction 3 has a larger IoU with the ground truth box than prediction 2, which can lead to better detection performance. However, when it comes to TEDS, prediction 3 cannot show any superiority compared to prediction 2, because both of them can cover the minimum box for table structure. When comparing prediction 2 and prediction 4, prediction 4 has a larger IoU with the ground truth box, making it better on detection performance, but it loses information of the row, making its performance in TEDS worse than prediction 2. Therefore, because of the definitions of COCO and TEDS metrics and the procedure of annotating datasets, a detection-based TSR model might be over-optimized towards detection performance without increasing the TEDS performance and sometimes can decrease the TEDS performance.

5.2.5 Rethinking Feature Extraction

As mentioned in Chapter 1 and Section 5.1, deformable convolution [27] has been applied in detection-based TSR [22, 86] and other related solutions [77, 165], demonstrating its effectiveness in improving detection performance. Deformable convolution uses a learnable grid offset to sample the grid points from the feature map and then apply the convolution operation to the sampled grid points, as defined by Equation 5.9,

$$\mathbf{z}_{p_0} = \sum_{p_n \in R} w(p_n) x(p_0 + p_n + \Delta p_n) \tag{5.9}$$

where p_0 is the location on the output feature map $\mathbf{z}$, p_n is the nth grid point in grid R, and Δp_n is the nth learnable offset. Since the offset Δp_n applied to the deformable

convolution is usually obtained by a regular convolution with small kernels, such as a 3 * 3 kernel, it can only improve the local feature instead of building long-range dependencies. However, building the long-range dependencies for the TSR task is important because of the characteristics of table components. More specifically, different parts of a single table component are often sparsely distributed across the table instead of a single area of compact pixels like common objects. Figure 5.6 shows a sample with its Row annotations. Taking the first Row as an example, as shown in Figure 5.6, it mainly contains three parts, which are distributed sparsely, and there is a large space between the first part and the second part, even though they all belong to a single target component. Therefore, it is important to build long-range dependencies together with improving local features, such as applying deformable convolution. Methods over-optimized local features, such as only applying deformable convolution, might degrade the performance of the TEDS.

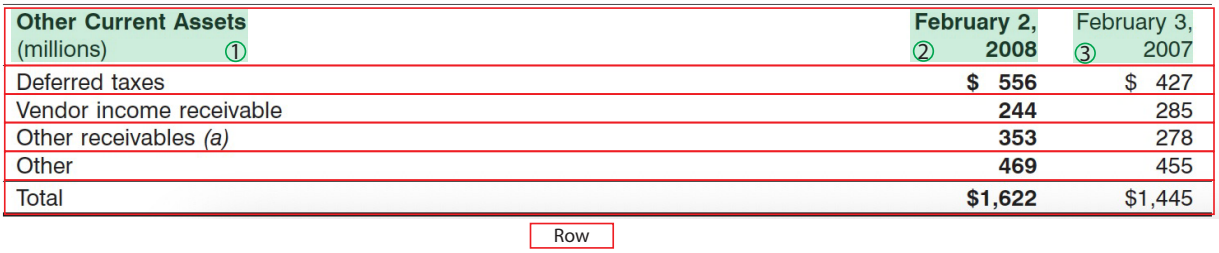


Figure 5.6: A sample from the FinTabNet dataset. Only Row annotations are showed for simplicity. The first Row in this Figure contains three major parts numbered 1 to 3.

5.3 Proposed Method

This section demonstrates how to fill the performance gap between detection-based and other types of TSR models by applying very simple methods to tailor the Cascade R-CNN model based on our analysis and findings in the previous sections. More specifically, this section introduces the refined problem formulation and gives the details of the proposed methods, including adjusting the parameters of the RPN, applying deformable convolution, and introducing the Spatial Attention Module.

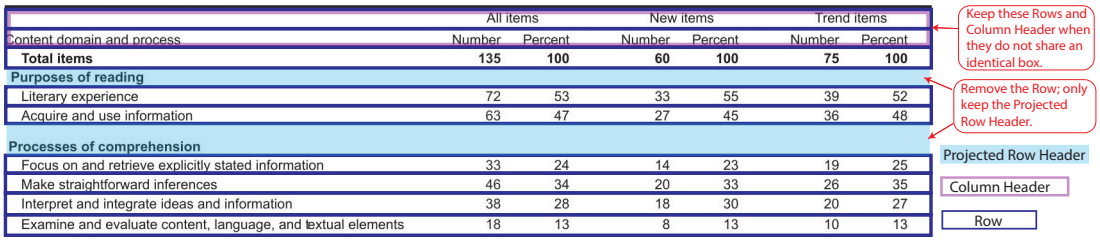
5.3.1 Proposed Problem Formulation

As mentioned in Sections 5.2.1 and 5.2.2, the definition of PubTables1M [24] can provide as much information as other types of solutions and is a multi-label detection problem, which is challenging for two-stage detectors. Therefore, we follow PubTables1M to define six table components: Table, Column, Row, Spanning Cell, Projected Row Header, and Column Header, and transform the formulation into a single-class detection problem. More specifically, Rows that share their bounding boxes with the Projected Row Header are removed, as shown in Figure 5.7a, and a Pseudo Class is used to replace the Rows and Column Headers when they share identical bounding boxes, as shown in Figure 5.7b. It is worth mentioning that only the Row, Projected Row Header, and Column Header are shown because the Table, Column, and Spanning Cell are the same as PubTables1M. These two samples are also in Figures 5.1b and 5.1c, which show their original definition in PubTables1M.

Formally, the ground truth Y in PubTable1M's definition for each image is a set of tuples containing bounding boxes and their corresponding labels, as defined by Equation 5.10, where b_i, c_i are the ith bounding box and its class, and values from 0 to 5 are the defined Table, Column, Row, Spanning Cell, Projected Row Header and Column Header, respectively.

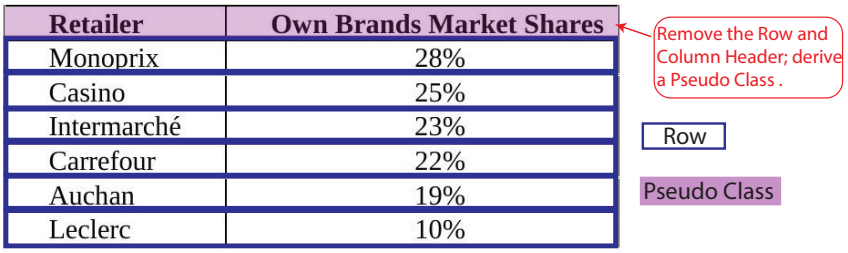
$$Y = \{(b_i, c_i)\}_{i=1}^{N}, c_i \in \{0, 1, 2, 3, 4, 5\}, \forall i \neq j, (c_i \neq c_j) \wedge ((b_i = b_j) \vee (b_i \neq b_j)) \tag{5.10}$$

By contrast, in this chapter, considering the observation that the defined Projected Row Headers are all Rows at the same time, only the Projected Row Headers samples are kept during the training. Since some Column Headers can share identical bounding boxes with corresponding Rows, a pseudo-class is derived for these overlapped samples and the original overlapped samples are removed. Therefore, during the training stage, the ground truth for each image is refactored to the regular single-label classification, as defined by Equation 5.11, where values 0 to 6 are the Table, Column, Row, Spanning Cell, Projected Row Header, Column Header and the Pseudo Class, respectively. During the testing stage, the results of Project Column Header are duplicated once to generate their corresponding prediction Rows, and the results of the pseudo-class are duplicated twice to generate the corresponding prediction Rows and Headers, so that the formulation defined by Equation 5.10 still can be followed to evaluate the model performance. Notably, this problem formulation is only applied to the tailored Cascade R-CNN model, and all other detection benchmark models are following the formulation of PubTables1M.



Content domain and process	All items Number	All items Percent	New items Number	New items Percent	Trend items Number	Trend items Percent
Total items	**135**	**100**	**60**	**100**	**75**	**100**
Purposes of reading						
Literary experience	72	53	33	55	39	52
Acquire and use information	63	47	27	45	36	48
Processes of comprehension						
Focus on and retrieve explicitly stated information	33	24	14	23	19	25
Make straightforward inferences	46	34	20	33	26	35
Interpret and integrate ideas and information	38	28	18	30	20	27
Examine and evaluate content, language, and textual elements	18	13	8	13	10	13

(a) An example of our problem formulation. In this example, two Rows are removed because their bounding boxes are identical to two Projected Row Headers.



Retailer	Own Brands Market Shares
Monoprix	28%
Casino	25%
Intermarché	23%
Carrefour	22%
Auchan	19%
Leclerc	10%

(b) An example of our problem formulation. In this example, a Pseudo Class is derived because its bounding box simultaneously belongs to a Row and a Column Header.

Figure 5.7: Examples of our proposed problem formulation. Since the definitions of Table, Column, and Spanning Cells are the same with PubTables1M, only Row, Column Header and Projected Row Header are shown for simplicity.

$$Y = \{(b_i, c_i)\}_{i=1}^{N}, c_i \in \{0, 1, 2, 3, 4, 5, 6\}, \forall i \neq j, (c_i \neq c_j) \wedge (b_i \neq b_j) \tag{5.11}$$

5.3.2 Tuning Parameters of RPN

As mentioned in Sections 5.2.1 and 5.2.3, regional proposal generation is a critical step in two-stage detectors, which need to be carefully considered for the TSR problem. Therefore, Aspect Ratios are adjusted, and the number of generated regional proposals is increased for the tailored model. More specifically, aspect ratios control the shape of the generated anchor boxes. Popular implementations of Cascade R-CNN, such as Detectron2 [150], usually use 0.5, 1.0, and 2.0 as default values, which can work well for detecting common objects, such as the objects in COCO [28] dataset. However, in the context of TSR, the range of aspect ratios is much larger because of the shape of the table components, as discussed in Section 5.2.3. Without proposing fancy new modules to select suitable values, the values are selected based on the statistics of the training sets. Taking the FinTabNet

dataset as an example, the aspect ratios of the defined components are shown in Figure 5.4. The maximum value is 140, far larger than the popular choices in common object detection. Besides, the majority of aspect ratios in Figure 5.4 are in the range between 1 and 60. Therefore, this parameter is extended for the proposed model as [0.0125, 0.025, 0.0625, 0.125, 0.25, 0.5, 1.0, 2.0, 4.0, 8.0, 16, 40, 80]. A further detailed parameter table is provided in Section 5.4. It is worth mentioning that when an aspect ratio is less than 1, its multiplicative inverse is applied to count the number of aspect ratios in Figure 5.4. This parameter is not further tuned through validation, which means that it might not be optimal. But this parameter has improved the model performance by around 2.7% as shown in Section 5.4.3. Besides, since increasing the number of proposals has been applied in existing studies [86] and demonstrated its effectiveness, it is increased for both the base Cascade R-CNN and our proposed model.

5.3.3 Spatial Attention and Deformable Convolution

As discussed in Section 5.2.5, building long-range dependencies for detecting the defined components is important. Inspired by the recent studies using large convolution kernels [166, 167], a Spatial Attention Module is introduced for the proposed solution, whose architecture is shown in Figure 5.8. For the design of the Spatial Attention Module, a similar architecture with MSCA [168] is used containing multiple branches and large kernel convolutions and spatial and depthwise separable convolution [169, 170] are used to reduce the number of parameters. More specifically, for the spatial separable convolution, a pair of $7*1$ and $1*7$ kernels is used to replace a typical $7*7$, use the pair of $11*1$ and $1*11$, and the pair of 21×1 and 1×21 is used to replace $11*11$ and $21*21$ kernels, respectively. For the depthwise separable convolution, the convolution is applied independently to each channel of the feature maps. Then, the outputs of the three branches are concatenated together as the input of a convolution layer with 1×1 kernel to make the channel dimension the same as the inputs. The proposed Spatial Attention Module can be easily inserted into the Backbone Network between two blocks because they do not change the feature shapes. For example, for a typical backbone network implemented by ResNet [152] containing a STEM block and four Residual Blocks, as shown in Figure 5.8, the Spatial Attention Module can be inserted after the last three Residual Blocks to generate the spatial attention, then the spatial attention can be applied to the original outputs of each Residual Block by Element-wise Multiplication. It is worth mentioning that the Spatial Attention Module shown in Figure 5.8 have independent trainable parameters, and all the feature maps are padded correspondingly to keep the size of the feature maps.

On the other hand, as discussed in Section 5.2.5, deformable convolution [27] can im-

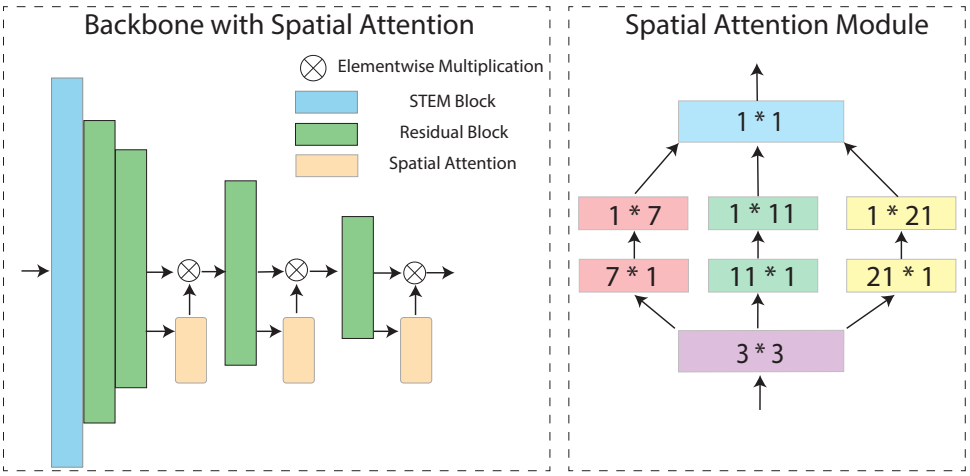


Figure 5.8: Architecture of proposed Spatial Attention Module. A ResNet backbone consists of a STEM Block and four stages of Residual Block. Our proposed Spatial Attention Module is inserted between the blocks of the backbone to build long dependencies.

prove the local feature generation and has been demonstrated to help improve the detection performance on the document image detection tasks by many studies [77, 86, 165]. Therefore, in this study, the proposed Spatial Attention Module and deformable convolution are applied to build long-range dependencies and improve local features together.

5.4 Experiments

5.4.1 Datasets and Experimental Settings

Four datasets are utilized to evaluate the proposed solution, including SciTSR [79], FinTabNet [81], PubTabNet [21] and PubTables1M [24]. As discussed in the study [171], FinTabNet and SciTSR datasets contain noise annotations that harm the model performance. Therefore, we use their cleaned versions proposed in the study in [171]. Each image sample in these four datasets contains only a table with extra padding pixels to ensure the entire table is extracted. The SicTSR dataset is collected from academic publications containing 7453, 1034, and 2134 samples for training, validation, and testing. PubTables1M dataset is a large-scale dataset for the TSR problem collected from the PMCOA corpus, containing 758849 training samples, 94959 validation samples, and 93834 testing

samples. Since the PubTabNet dataset does not provide original PDF files, it is impossible to process it to make detection annotations. Besides, its testing is not publicly available. Therefore, its validation set is used to evaluate the model trained with the PubTable1M dataset. Following the study in [171], the code base in [172] is used to process the datasets and align the formats of these datasets. FinTabNet is also a large dataset widely used for the TSR problem, containing 78537, 9289, and 9650 samples for training, testing, and validation. FinTabNet is collected from the annual reports of companies, making its data source different from the other datasets. Table 5.1 summarizes the datasets used in this study for the model evaluation.

Table 5.1: Summary of datasets.

Dataset	Train	Validation	Test
SciTSR [79]	7,453	1,034	2,134
FinTabNet [81]	78,537	9,650	9,289
PubTabNet [21]	500,777	9,115	-
PubTables1M [24]	758,849	94,959	93,834

Since the TSR problem in this study is formulated as an object detection problem, we use both detection and cell-level TSR metrics for the model evaluation. For the detection metric, the widely accepted COCO metrics [28] are employed, which has been discussed in Section 5.2.4. More specifically, mean Average Precision (mAP), $AP50$, $AP75$, AP_s, AP_m, AP_l, and object-specific AP scores are used as metrics, where $AP50$, $AP75$ are the APs using 0.50 and 0.75 as IoU thresholds, respectively. AP_s, AP_m, and AP_l are the APs of different target object sizes, defined by Equation 5.12.

$$object\_size = \begin{cases} small & \text{if area } <32^2 \text{ px} \\ medium & \text{if } 32^2 <\text{area } <64^2 \text{ px} \\ large & \text{otherwise} \end{cases} \tag{5.12}$$

For the TSR metric, structure-only Tree-Edit-Distance-Based Similarity(TEDS) [21] is used, which is firstly introduced in the study to overcome the drawbacks of adjacency relation metrics and can be defined as Equation 5.8 as discussed in Section 5.2.4. Structure-only TEDS only considers the HTML tags without extracting their contents to avoid the influence of OCR tools. The testing samples can be categorized into simple and complex groups based on whether they have cells spanning multiple columns and rows. Notably, for the evaluation of detection performance, the formulation defined in Equation 5.10 is used, which is also the problem definition of PubTables1M [24], and the results generated by

our single-label detection can be easily transformed into the multi-label detection results defined by PubTables1M [24], as discussed in Section 5.3.1.

5.4.2 Implementation Details and Experimental Results

To verify the effectiveness of our proposed solution, all three types of state-of-the-art methods are included, as discussed in Chapter 2. For the detection-based methods, Cascade R-CNN [25], Deformable-DETR [61] and Sparse R-CNN [10] are used as benchmark models, in which Cascade R-CNN [25] is also the based model of the proposed methods, Deformable-DETR and Sparse R-CNN are two state-of-the-art transformer-based detection models. For the image-to-sequence models, EDD [21], TableFormer [173], TableMaster [17], VAST [85] and MTL-TabNet [16] are included. TSRFormer-DQ-DETR [93] and RobustTabNet [94] are two state-of-the-art models following the pipeline of detecting separation lines and then merging cell grids, which can be treated as a graph-based model as discussed in Chapter 2. TSRNet [88] is also a graph-based methods which detect table cells first, then applies GNN to build the relations among the detected cells.

Table 5.2: Key training parameters of the proposed model. MAX_ITER and STEPS are for the FinTabNet dataset as examples.

Parameter	Value	Description
RESNETS.NORM	nnSyncBN	Batch Normalization for the Backbone Network
MAX_ITER	112,500	total number of mini-batch
STEPS	84,375	the mini-batch to apply the learning rate schedule
SCHEDULER	MultiStepLR	the scheduler to change the learning rate
NMS_THRESH	0.9	non-maximum suppression threshold
PRE_NMS_TOPK_TRAIN	4000	RPN proposals to keep before applying NMS in training
PRE_NMS_TOPK_TEST	2000	RPN proposals to keep before applying NMS in testing
POST_NMS_TOPK_TRAIN	4000	RPN proposals to keep after applying NMS in training
POST_NMS_TOPK_TEST	2000	RPN proposals to keep after applying NMS in testing
DEFORM_ON_PER_STAGE	[True, True, True, True]	whether to use deformable convolution in backbone stages

Cascade R-CNN and the proposed method are implemented based on the Detection2 [150], the Deformable-DETR is based on detrex [174], and the Sparse R-CNN is based on its official codebase. The default parameters of Deformable-DETR and Sparse R-CNN are used. For the Cascade R-CNN baseline, the number of regional proposals and the batch normalization method are aligned to the proposed solution, as shown in Table 5.2. All these detection models are using ResNet50 [152] pre-trained with ImageNet [153] as the backbone network. TableMaster [17] is also re-trained with the FinTabNet dataset based on its official code base. The proposed method is termed with TSRDet for fast reference. For the

implementation of the proposed TSRDet, aspect ratios in the anchor box generation are set as [0.0125, 0.025, 0.0625, 0.125, 0.25, 0.5, 1.0, 2.0, 4.0, 8.0, 16, 40, 80], and other key parameters are summarized in Table 5.2. Notably, to calculate the structure-only TEDS, the scripts provided by study in [172] are used to generate the HTML sequences from the detected components and all benchmark models, except the proposed model, using the original definition of PubTables1M, which treats all table components independently with its multi-label detection setting. All the models are trained with 240, 120, and 60 epochs for the SciTSR, FinTabNet, and PubTables1M datasets, respectively. For the training of all models, we use the validation set for the model selection and hyperparameter tuning.

Table 5.3: Experimental results on SciTSR dataset with structure-only TEDS score. *Sim.* means the tables without spanning cells and *Com.* represents the tables with spanning cells.

Model	TEDS-struc.(%)		
	Sim.	Com.	All
Cascade R-CNN	77.31	84.74	79.09
Deformable-DETR	98.17	94.59	97.30
Sparse R-CNN	**99.08**	95.92	98.30
TSRDet(Proposed)	98.59	**97.88**	**98.41**

Table 5.4: Experimental results on FinTabNet dataset with structure-only TEDS score. *Sim.* means the tables without spanning cells and *Com.* represents the tables with spanning cells.

Model	TEDS-struc.(%)		
	Sim.	Com.	All
EDD [21]	88.40	92.08	90.60
TableFormer [173]	97.50	96.00	96.80
TableMaster [17]	98.36	98.28	98.32
VAST [85]	-	-	98.63
MTL-TabNet [16]	99.07	98.46	98.79
TSRFormer-DQ-DETR [93]	-	-	98.40
Cascade R-CNN	82.17	92.50	87.49
Deformable-DETR	98.08	97.54	97.81
Sparse R-CNN	98.36	97.91	98.13
TSRDet(Proposed)	**99.08**	**99.02**	**99.05**

The experimental results regarding the structure-only TEDS and COCO metrics are shown in Tables 5.3, 5.4, 5.5, 5.6 and 5.7, which can demonstrate the superiority of the proposed solution. For the SciTSR dataset, the proposed TSRDet can improve the baseline Cascade R-CNN by 19.32% regarding the structure-only TEDS, outperforming Deformable-DETR and Sparse R-CNN. When it comes to COCO metrics, the mAP of the proposed TSR is as good as Deformable-DETR, outperforming other benchmark models. Similarly, the proposed TSRDet can also outperform benchmark models regarding both COCO metrics and structure-only TEDS on the FinTabNet and PubTables1M datasets. It is worth mentioning that PubTabNet dataset does not provide original PDF files, making it hard to generate detection annotations. Therefore, the model performance reported in Table 5.6 is calculated using the model trained with the PubTable1M dataset. Although the PubTable1M and PubTabNet datasets have misalignments regarding the ground truth HTML sequences, the proposed method still shows competitive performance compared with other state-of-the-art methods, as shown in Table 5.6. Figure 5.9 shows a prediction sample and its generated HTML sequence after post-processing, demonstrating the capacities of the proposed solution.

Table 5.5: Experimental results on PubTables1M dataset with structure-only TEDS score. *Sim.* means the tables without spanning cells and *Com.* represents the tables with spanning cells.

Model	TEDS-struc.(%)		
	Sim.	Com.	All
Cascade R-CNN	82.73	85.21	83.78
Deformable-DETR	97.54	93.14	95.73
Sparse R-CNN	99.04	95.90	97.72
TSRDet(Proposed)	**99.19**	**97.66**	**98.55**

5.4.3 Ablation Study

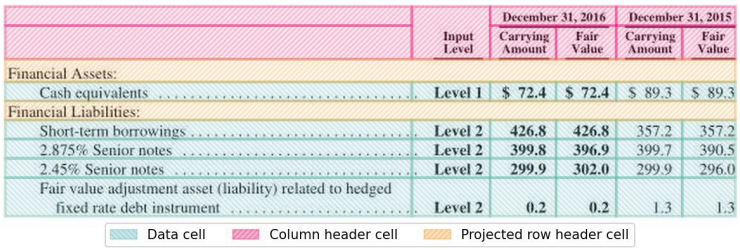
In this section, experiments are conducted on the FinTabNet dataset to demonstrate the effectiveness of the applied methods, including using the proposed single-label detection formulation, tuning parameters of RPN, and applying the deformable convolution and spatial attention. It is worth mentioning that tuning RPN parameters includes increasing the number of proposals and adjusting the aspect ratios. Since other studies have successfully applied the effectiveness of increasing the number of proposals, it is applied to both the Cascade R-CNN baseline and the proposed TSRDet, as discussed in Section 5.2.3 and 5.4.2.

Table 5.6: Experimental results on PubTabNet validation set with structure-only TEDS score. *Sim.* means the tables without spanning cells and *Com.* represents the tables with spanning cells. The proposed model is trained with PubTable1M dataset, while the benchmark models are trained with PubTabNet dataset.

Model	TEDS-struc.(%)		
	Sim.	Com.	All
EDD [21]	91.10	88.70	89.90
RobustTabNet [94]	-	-	97.00
TSRNet [88]	-	-	95.64
VAST [85]	-	-	97.23
TableFormer [173]	98.50	95.00	96.75
MTL-TabNet [16]	99.05	96.66	97.88
TSRDet(Propsed)	96.99	94.99	96.58

Table 5.7: Experimental results with Mean Average Precision (mAP).

Dataset	Model	mAP	$AP50$	$AP75$	AP_s	AP_m	AP_l	Table	Column	Row	Spanning Cell	Projected Row Header	Column Header
SciTSR	Cascade R-CNN	93.89	95.27	94.80	95.81	93.89	92.96	98.96	98.63	96.33	88.58	83.80	97.01
	Deformable-DETR	96.28	97.39	97.01	96.75	96.55	96.07	98.96	98.63	97.26	93.84	90.86	98.15
	Sparse R-CNN	94.78	96.17	95.48	95.49	95.07	90.08	98.98	98.30	97.93	88.06	86.92	98.49
	TSRDet(Proposed)	96.28	96.79	96.57	99.01	96.42	95.65	98.97	99.25	98.57	95.30	87.06	98.50
FinTabNet	Cascade R-CNN	95.23	97.53	96.90	87.32	95.31	93.08	99.00	96.69	96.96	84.43	96.63	97.64
	Deformable-DETR	96.68	98.42	97.98	75.17	95.53	95.58	99.00	97.55	96.95	91.91	96.62	98.04
	Sparse R-CNN	96.38	98.37	97.69	62.11	96.22	95.86	99.01	97.79	97.84	88.39	97.29	97.97
	TSRDet(Proposed)	97.50	98.33	98.09	91.60	97.40	97.15	99.01	98.83	97.99	94.62	96.61	97.93
PubTables1M	Cascade R-CNN	93.40	95.38	94.76	85.75	93.32	92.57	99.01	98.76	87.56	82.18	95.81	97.11
	Deformable-DETR	94.82	97.43	96.79	78.33	92.55	94.48	98.99	97.89	95.84	85.04	95.43	95.74
	Sparse R-CNN	96.46	98.14	97.60	84.25	95.73	96.45	99.00	98.42	98.03	87.85	97.91	97.57
	TSRDet(Proposed)	97.72	98.26	98.04	94.76	97.43	97.33	99.01	98.99	98.41	94.21	97.88	97.85



(a) A prediction sample after post-processing.

```
<table>
  <thead>
      <th></th>
      <th rowspan="2"></th>
      <th colspan="2"></th>
      <th colspan="2"></th>
  </thead>
  <thead>
      <th></th>
      <th></th>
      <th></th>
      <th></th>
      <th></th>
  </thead>
  <tr><td colspan="6"></td></tr>
  <tr>
      <td></td> <td></td> <td></td> <td></td><td></td> <td></td>
  </tr>
  <tr><td colspan="6"></td></tr>
  <tr>
      <td></td><td></td><td></td><td></td><td></td><td></td><td></td>
  </tr>
  <tr>
      <td></td><td></td><td></td><td></td><td></td><td></td><td></td>
  </tr>
  <tr>
      <td></td><td></td><td></td><td></td><td></td><td></td><td></td>
  </tr>
  <tr>
      <td></td><td></td><td></td><td></td><td></td><td></td><td></td>
  </tr>
</table>
```

(b) The generated HTML sequence after post-processing.

Figure 5.9: A sample of prediction result from the FinTabNet testing set.

Therefore, only the impact of adjusting the aspect ratios for tuning parameters of RPN is discussed in this section.

The experimental results are shown in Tables 5.8 and 5.9, in which Asp_Ratio Tuning, Single_Label, DEFORM, and S_Attn are shorts for applying aspect ratio tuning, single label formulation, deformable convolution, and spatial attention, respectively. Even though Cascade R-CNN baseline can reach 95.06% regarding the mAP, its overall structure-only TEDS only reaches 82.70%. After tuning the aspect ratios for the anchor generation, the structure-only TEDS is increased to 90.23%, even though the mAP is only increased from 95.06% to 95.54%. Applying deformable convolution without other methods can improve the detection performance significantly but lead to a worse structure-only TEDS if we compare Ablation 1 and the Cascade R-CNN baseline. Ablation 3 and Ablation 4 show that transforming the multi-label detection formulation into a single-label formulation can significantly improve the performance and also make deformable convolution improve the model performance. Applying both deformable convolution and spatial attention together can further improve the model performance from the results of Abliation 4, 5 and TSRDet, as shown in Table 5.8. On the other hand, when it comes to detection metrics, applying deformable convolution always brings performance improvements from the results of Ablation 1 and Ablation 4, which can verify our analysis on the mismatch of detection metrics and cell-level metrics in Section 5.2.4.

Table 5.8: Ablation study results on FinTabNet dataset with structure-only TEDS score. Asp_Ratio Tuning, Single_Label, DEFORM, and S_Attn are shorts for applying aspect ratio tuning, single-label formulation, deformable convolution, and spatial attention.

Model	Asp_Ratio Tuning	Single_Label	DEFORM	S_Attn	TEDS-struc.(%)		
					Sim.	Com.	All
Cascade R-CNN					82.17	92.50	87.49
Ablation 1			✓		81.45	87.11	84.35
Ablation 2	✓				84.27	95.80	90.23
Ablation 3	✓	✓			95.17	98.63	96.95
Ablation 4	✓	✓	✓		96.44	99.14	97.83
Ablation 5	✓	✓		✓	96.95	98.75	97.88
TSRDet(Proposed)	✓	✓	✓	✓	99.08	99.02	99.05

Table 5.9: Ablation study results regarding mean Average Precision (mAP). The model names are aligned with models in Table 5.8.

Model	mAP	$AP50$	$AP75$	AP_s	AP_m	AP_l	Table	Column	Row	Spanning Cell	Projected Row Header	Column Header
Cascade R-CNN	95.23	97.53	96.90	87.32	95.31	93.08	99.00	96.69	96.96	84.43	96.63	97.64
Ablation 1	97.22	98.03	97.90	90.11	96.72	96.76	99.00	98.95	96.16	94.98	96.01	98.19
Ablation 2	95.54	97.54	96.91	87.43	95.79	94.04	99.00	97.04	97.64	84.84	96.67	98.02
Ablation 3	95.51	97.56	96.94	88.43	95.52	93.76	99.00	97.31	97.87	84.74	96.82	97.28
Ablation 4	97.83	98.37	98.13	91.91	97.65	97.58	99.00	98.96	98.33	95.78	96.98	97.93
Ablation 5	96.97	97.84	97.58	90.32	96.88	96.21	99.00	98.83	98.03	91.97	96.58	97.37
TSRDet(Proposed)	97.50	98.33	98.09	91.60	97.40	97.15	99.01	98.83	97.99	94.62	96.61	97.93

5.5 Discussions and Analysis

Sections 5.4.2 and 5.4.3 have demonstrated the effectiveness of the proposed solution. This section further discusses some observations from the experimental results and how these observations verify the analysis in Section 5.2.

5.5.1 Multi-label Detection

As discussed in Sections 5.2.1 and 5.2.2, multi-label detection tasks are difficult for two-stage detection models, but transformer-based detection models with learnable proposals can deal with multi-label detection tasks. Besides, the problem formulation of PubTables1M is a multi-label task, making it difficult for two-stage detection models. The experimental results from sections 5.4 can demonstrate this analysis. For example, as shown in Table 5.3, the performance of Deformable-DETR and Sparse R-CNN are 97.81% and 98.13% regarding the structure-only TEDS, which are very close to the performance of proposed TSRDet (98.41%) and far better than the Cascade R-CNN baseline (79.09%). Notably, as mentioned in Section 5.4, all the models, except the proposed TSRDet, are using the multi-label detection setting. Therefore, two transformer-based detection models show promising results in the multi-label detection setting. Similarly, the experiments on the FinTabNet and PubTables1M datasets also show similar results. For example, the structure-only TEDS performance of Sparse R-CNN, TSRDet, and Cascade R-CNN baseline are 97.72%, 98.55%, and 83.78% on the PubTables1M dataset, 98.13, 99.05, and 87.49 on the FinTabNet dataset.

5.5.2 The Misalignment of Metrics

The experimental results in Sections 5.4.2 and 5.4.3 show the misalignment of COCO and TEDS metrics many times. For example, in Table 5.7, both the Deformable-DETR and

Other Current Assets (millions)	February 2, 2008	February 3, 2007
Deferred taxes	$ 556	$ 427
Vendor income receivable	244	285
Other receivables *(a)*	353	278
Other	469	455
Total	$1,622	$1,445

(a) A sample prediction result of the Ablation 1 model. The mAP and structure-only TEDS are 97.22 and 84.35, respectively. We only include the columns' predictions for simplicity.

Other Current Assets (millions)	February 2, 2008	February 3, 2007
Deferred taxes	$ 556	$ 427
Vendor income receivable	244	285
Other receivables *(a)*	353	278
Other	469	455
Total	$1,622	$1,445

(b) A sample prediction result of the Ablation 3 model. The mAP and structure-only TEDS are 95.51 and 96.95, respectively. We only include the columns' predictions for simplicity.

Figure 5.10: Comparison of results from Ablation1 and Ablation3 models. Even though Ablation 1 can achieve better detection performance, its performance regarding structure-only TEDS is much lower than that of Ablation 3 model.

our proposed TSRDet can reach 96.28% regarding mAP on the SciTSR dataset, which is better than that of Sparse R-CNN. However, when it comes to structure-only TEDS, as shown in Table 5.3, both TSRDet and Sparse R-CNN can perform better than Deformable-DETR. Similar results also appear in the experiments on the FinTabNet dataset. As shown in Table 5.7, the mAP of Sparse-RCNN and Deformable-DETR are 96.38% and 96.68%, while their structure-only TEDS are 97.81% and 98.13%. More similar results can be found in the results of the ablation study, such as Ablation 1 and Ablation 3, as shown in Tables 5.8 and 5.9. To further verify the discussion in Section 5.2.4, prediction results of Ablation 1 and 3 are shown in Figure 5.10. As discussed in Section 5.2.4, COCO metrics are relied on IoU scores, while TEDS is not. Therefore, as shown in Figure 5.10a, Ablation 1 with deformable convolution can better fit the extra white areas to improve the detection performance, but it cannot improve the TEDS compared with Ablation 3 whose result is shown in Figure 5.10b. It is worth mentioning that the ground truth of the sample in Figure 5.10 has been shown in Figure 5.5.

5.5.3 Deformable Convolution and Spatial Attention

As discussed in Sections 5.2.4 and 5.2.5, both generating good local features and building long-range dependencies are essential for a detection-based TSR model, and deformable convolution can improve the local feature generation but has the risk leading to the over-optimization to the detection performance. The ablation study's experimental results can somewhat demonstrate our analysis. Considering the performance of Ablation 1 with deformable convolution in Tables 5.8 and 5.9, its TEDS is 84.35%, lower than the Cascade R-CNN baseline (87.49%), but its mAP is improved from 95.23% to 97.22%. These results not only show the misalignment of COCO and TEDS metrics but also demonstrate that merely improving local features can make the model fit empty spaces better, as shown in Figures 5.10a, but does not help alleviate the multi-label detection issue. Therefore, deformable convolution needs to be applied with other methods. On the other hand, the proposed Spatial Attention Module can improve the mAP and structure-only TEDS simultaneously, comparing the performance of Ablation 3 and 4, and also can be used with deformable convolution together to improve the structure-only TEDS further, as shown in Figure 5.8, demonstrating the effectiveness of building long-range dependencies.

5.5.4 Other Observations

Besides the observations discussed in previous sections, the experimental results also show other phenomena that can be helpful in the model design. One observation is that Cascade R-CNN has better detection performance on small objects than Sparse R-CNN. For example, on the FinTabNet dataset, APs of Sparse R-CNN is only 62.11%, while the Cascade R-CNN baseline and our proposed TSRDet reach 87.32% and 91.60%. This phenomenon might be caused by their methods of generating regional proposals. As discussed in Section 5.2.1, Cascade R-CNN uses RPN to generate regional proposals, which regress and classify anchor boxes, and the anchor boxes are generated by sliding the pre-defined boxes with different aspect ratios and sizes on the feature map of multiple scales. Therefore, Cascade R-CNN uses a dense proposal generation method [10] with more region proposals, meaning that Cascade R-CNN can use the parameters of RPN to generate more high-quality small region proposals. By contrast, Sparse R-CNN uses sparse learnable regional proposals to replace dense proposals generated by the RPN, which can avoid parameter tuning of RPN but limit its performance on small objects. Another interesting observation is that the baseline Cascade R-CNN can work better on complex tables than simple tables, which is very different from other benchmark models. This phenomenon is caused by the fact that the spanning cells in complex tables are usually in the Column Row

Headers, which can alleviate the multi-label detection issue. For example, Figures 5.1b and Figures 5.1c show two samples from PubTables1M dataset, in which the former is a complex table and the latter is a simple table. Because of the existence of Spanning Cells in Figure 5.1b, the Column Header does not share its bounding box with any rows, which avoids multi-label detection. By contrast, the sample in Figure 5.1c does not contain any Spanning Cell, making its Column Header share its bounding box with a Row, which is the challenging multi-label detection to Cascade R-CNN. As comparisons, Deforamble-DETR and Sparse R-CNN can deal with multi-label detection, and their performance on simple tables is better than complex tables regarding the structure-only TEDS, as shown in Tables 5.3, 5.4and 5.5.

5.5.5 Summary of Insights

This section summarizes the key insights and critical design aspects for a detection-based TSR model. It is worth mentioning that the rationale behind these insights, along with the experiments validating them, has been thoroughly discussed in Sections 5.2, 5.4, and 5.5. First, a detection-based TSR solution needs to define the target table components properly to provide full table structural information. Some studies [22, 23, 86] over-simply the target detection components without Headers and Projected Row Headers, making them unable to fully recover the complex table structures. Furthermore, the problem formulation should align with the capacities of the employed detection model. For instance, PubTable1M [24] defines six types of target table components to fully reconstruct the complex table structures. However, it presents a multi-label detection definition, posing challenges for two-stage detection models. Therefore, this thesis further develops the formulation of PubTable1M by introducing a pseudo-class to transform multi-label detection to regular single-label detection, as discussed in Section 5.3.1. Thirdly, existing studies usually employ COCO metrics to evaluate the detection-based TSR models. However, COCO metrics are insufficient for evaluating the TSR models because ground truth boxes often exceed the minimum bounding boxes required for capturing table structures, as discussed in Section 5.2.4, and models may optimize towards accommodating easier components with additional spaces, rather than effectively identifying challenging components. Hence, this thesis incorporates structure-only TEDS for model evaluation and introduces a Spatial Attention Module. The module is designed to establish long-range dependencies, enhancing the model's ability to explore and address challenging components effectively. Fourthly, two-stage and transformer-based detection models have different capacities in the context of the TSR task. This thesis leverages Cascade R-CNN and Sparse R-CNN as illustrative examples to highlight their differing capacities. More specifically, Sparse R-CNN excels

in handling multi-label detection tasks without the need for tuning the region proposals because of its utilization of sparse learnable proposals, as discussed in Sections 5.2.1 and 5.2.3. By contrast, Cascade R-CNN cannot deal with multi-label detection tasks and needs to carefully tune the parameters of proposal generation because of the aspect ratios of defined table components, as discussed in Section 5.2.1, 5.2.2 and 5.2.3. Additionally, Cascade R-CNN demonstrates superior performance on small objects compared to Sparse R-CNN, partially attributed to its dense and tunable proposal generation, as illustrated in Section 5.4.2. At last, while enhancing local feature extraction, such as employing deformable convolution, often leads to improved detection performance, it may not necessarily translate to enhanced TSR performance. It is necessary to build long-range dependencies, as discussed in Section 5.2.5. To sum up, it is imperative to ensure proper alignment between the problem formulation, capacities of detection models, evaluation metrics, and feature extraction in the context of a detection-based TSR solution. Our proposed Cascade R-CNN can be a demonstrative application of these insights in designing an effective detection-based TSR model.

5.6 Summary of the Chapter

This chapter first revisits existing detection-based TSR solutions and analyzes the critical design aspects for a successful detection-based TSR model, including the problem formulation, the characteristics of detection models, and the characteristics of TSR tasks. The analysis can be a guideline for improving the performance of a detection-based model. To demonstrate the analysis and findings, TSRDet is proposed by applying simple methods to tailor the Cascade R-CNN, which can outperform different types of state-of-the-art models, including image-to-sequence and graph-based models. Even though only very simple methods are applied to a two-stage detection model, there should be other methods to improve the model further based on our analysis. For example, vision transformers can be considered for building long-range dependencies. Transformer-based detection models, such as Sparse R-CNN, can also be considered as base models with the benefits of dealing with multi-label detection tasks and learnable proposals. Besides, since the proposed method is detection-based and focuses on well-formatted, visually rich documents, one major limitation is that it may fail to deal with irregular tables, such as rotated and distorted tables. Integrating instance segmentation with detection models or applying graph-based approaches can be further directions for dealing with irregular tables.

Chapter 6

Table Question Answering

Question Answering (QA) is a widely discussed problem formulation aiming to answer the query question based on the given context information, where the context information can be plain text [175–177], knowledge base [178–180], images [181], videos [182] and other sources. Since tables are widely used to summarize critical information in many data sources, such as visually rich documents and web pages, Table Question Answering (Table-QA) [103, 104, 130, 183, 184] has recently drawn much research attention. Typically, tables can be easily categorized into two groups: structured and semi-structured tables. Structured tables are usually from relational database systems with explicit schema describing their structures and data types, meaning the programming languages, such as SQL, can naturally process them. By contrast, tables from other sources, such as web pages and visually rich documents, are usually semi-structured without schema requirements, resulting in complex structures and heterogeneous data types, making the QA task on these semi-structured tables more challenging. This chapter focuses on the Table-QA problem on the semi-structured tables, which is a more challenging setting.

6.1 Motivation

As discussed in Chapter 1, table question answering is the last step of the proposed first solution in this thesis, taking TSR models' outputs and corresponding questions as inputs. As tables from visually rich documents are usually images, this focuses on semi-structured tables, which can be obtained by applying the proposed TD and TSR methods. Since the limitations of LLMs, such as their capacities for arithmetic calculation and reasoning, applying programming languages such as Python and SQL is a popular choice. However,

semi-structured tables can contain complex structures and inconsistent data types and sometimes can have numerous columns and rows, making it challenging to prompt LLMs to generate runnable programming codes. Besides, the performance of LLMs heavily relies on the prompts and the demonstrations when ICL is applied, and how to craft and select demonstrations remains an open issue. Therefore, this thesis defines a series of atomic operations that can guide the demonstration, crafting, and selection. Besides, a multi-stage prompting method is proposed to mitigate the issues caused by reasoning errors and inconsistent data types. To sum up, this chapter covers the following aspects:

1. This chapter introduces a series of atomic operations to describe and measure the similarities of QA tasks. The defined operations can be a guide to craft demonstrations and a distance metric for demonstration selection.

2. A three-stage prompting method is proposed, including task-planning, task-conducting and task-correction stages, which can alleviate the issues caused by the complex structures, heterogeneous data types, huge tables and the limitations of LLMs and reduce the inference cost.

3. Comprehensive experiments are conducted on WikiTableQA and TabFact datasets with open-source language models. The experimental results demonstrate that our proposed method can outperform the baseline models by 4% to 23% in various experiment settings, achieving state-of-the-art performance. Our experimental results can also be the benchmark for open-source LLMs for the Table-QA problem, as most current studies rely on commercial LLMs.

The rest of this chapter is organized as follows: Section 6.2 describes the proposed method for the Table-QA task. Section 6.3 shows the experimental results and discusses various aspects of the proposed method. At last, Section 6.5 summarizes the key points of this chapter.

6.2 Proposed Method for Table-QA

6.2.1 Overall Workflow

As discussed in Chapter 1, this chapter explores prompting open-source LLMs to generate Python code for the semi-structured Table-QA problem to mitigate the issues caused by

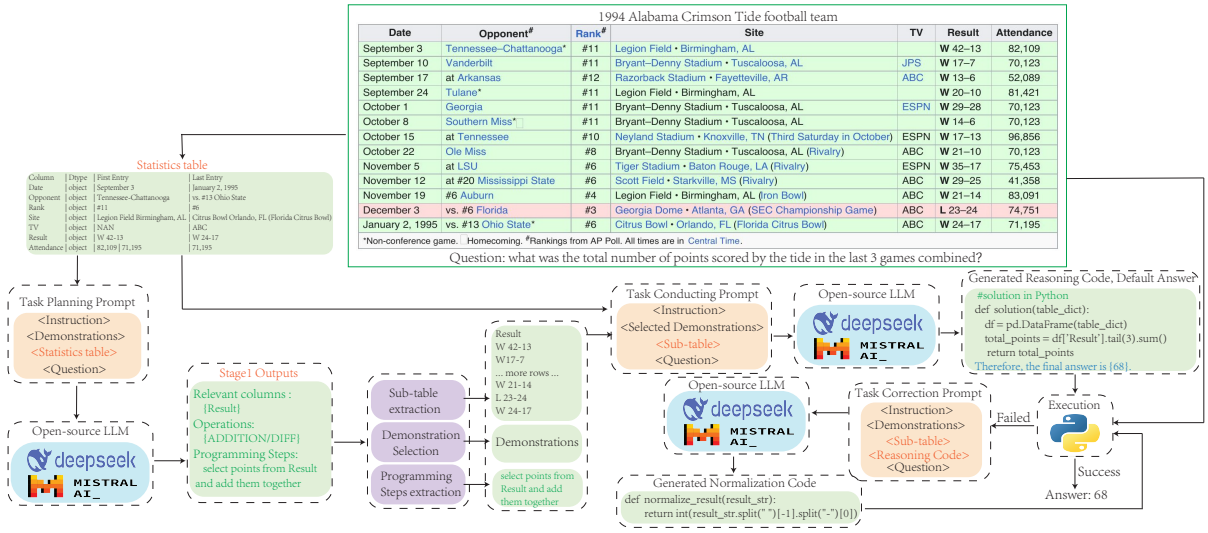


Figure 6.1: The workflow of the proposed solution.

complex table structures, heterogeneous data types, huge tables, and limitations of LLMs. Specifically, we propose a three-stage prompting method, including task-planning, task-conducting and task-correction stages. The proposed workflow is shown in Figure 6.1, in which the question is *the total number of points scored by the tide in the last 3 games combined*. To answer this question, the statistics table of the given table contains information regarding the column names, data types, and the first and last entries, which are first generated as a part of the task-planning prompt, together with instructions, demonstrations, and questions, as shown in Figure 6.2. Then, the task-planning prompt is fed into the open-source LLM to make the reasoning plan, including Relevant Columns, Operations, and Programming Steps. It is worth mentioning that the statistics table can be far more compact than the original table when the original table is huge, which can reduce the number of prompt tokens. With the Relevant Columns, Operations and Programming Steps from the results of the task-planning step, the Relevant Columns are used to extract the contents of these columns, the Operations are used to select the demonstrations, and the Programming Steps are parts of the instructions to guide the reasoning function generation. With these processed results, the task-conducting prompt is constructed and fed into the open-source LLM to generate the reasoning Python function and a default answer, in which the default is treated as the final answer if the code fails to run. Notably, since we use the Pandas library in our solution, the original table needs to be represented as a dictionary of the list and fed into the generated function as a parameter. As running the

generated Python code is almost cost-free compared with LLM inference, and some tables do not need to be normalized, we try to execute the reasoning function first. If there is no error from the execution, then the result of the execution should be the final result. However, if the execution fails, we construct the task-correction prompt containing the content of Relevant columns and reasoning code to generate the normalization function, apply the normalization to the original table, and then feed the normalized original table as the parameter to the reasoning function to run it again. For the example in Figure 6.1, we need to extract the string "W 21-14", "L 23-24" and "W 24-17" first and then extract the scores "21", "23", "24" and convert them into integers, which beyond the capacities of the LLM. Therefore, the first run of the reasoning function would fail, even though its reasoning logic is correct. While the normalization function can correctly extract and convert the points from the string to integers, the second run of the reasoning function should be successful.

6.2.2 Demonstration Crafting and Selection

As shown in Figure 6.1, three prompts need to be constructed to answer a question; the demonstrations used in these three prompts are critical for the LLM's performance, especially for the task-conducting stage to generate the reasoning function. Therefore, we defined a series of atomic operations to describe the reasoning logic to answer questions, as shown in Table 6.1. For the task-planning step, we craft a question-and-answer pair for each type of operation and instruct the LLM to generate the Relevant Columns, Operations and Programming Steps, as shown in Figure 6.2. For the task-conducting prompts, we construct two question and reasoning function pairs for each defined operation and use operation as the metric to select these pairs. Even though we prompt the LLM to select one defined operation as output, the LLM can generate results without following the instructions. Therefore, when the Operations cannot be matched, the default setting contains all the question and function pairs. Figure 6.3 shows an example of COUNT operation, which includes a Meta Information table, Column Details and two questions with their Python solutions and default answers.

You defined 5 types of operations to answer the questions based on the given table. The operations are defined as follows:
1. SelectTable: select the content from the given table based on some criteria
2. ADDITION/DIFF: addition or subtraction
3. AVG: average of several numbers
4. COUNT: count the number of spans
5. MAX/MIN: select the maximum/minimum one from given numbers
To answer the questions, one of these operations might be needed. You are going to analyze two tables. Select one operation from defined operations as needed Operation and also give Programming Steps and Revelant Columns to the questions.
Read the table below regarding "2008 Clasica de San Sebastian" to give the needed operations, reasoning steps and revelant columns to the following questions.
Column | Dtype | First Entry | Last Entry
Rank | int64 | 1 | 10
Cyclist | object | Alejandro Valverde (ESP) | David Moncoutie (FRA)
Team | object | Caisse d' Epargne | Cofidis
Time | object | 5h 29' 10 | + 2
UCI ProTour Points | int64 | 40 | 1
Question: who ranked at the 7th place?
Programming Steps: Column Rank contains the ranking information, column Cyclist contains the names of cyclists. To answer this question, select cyclist from column Cyclist ranked at the 7th place based on column Rank.
Operations:{SelectTable}.
Revelant Columns: {Rank,Cyclist}.

...More Lines...

Question: What is the average points of all cyclists listed in the table?
Programming Steps: Columns UCI ProTour Points contains the points information. To answer this question, calculate the mean value of all points from UCI ProTour Points
Operations:{AVG}.
Revelant Columns: {UCI ProTour Points}.
Read the table blow regarding " + <title> + " to answer the following one question:
<Statistics Table>
<Question>

Figure 6.2: The task-planning prompt. The defined operations and the statistics table are highlighted in green and yellow. <title>, <Statistics Table> and <Question> are from the table to be analyzed.

Table 6.1: Defined operations for Demonstration selection for code generation.

Operation Type	Operation Name	Description
Reasoning	SelectTable	select a cell from the table based on a criteria
	ADDITION/DIFF	addition or subtraction
	TIMES/DIVISION	production or quotient of two numbers
	AVG	average of several numbers
	COUNT	count the number based on a criteria
	MAX/MIN	select the maximum/minimum one from given numbers
	ARGMAX/ARGMIN	select key with highest/lowest value from key-value pairs

You have the following 2 tables' meta information and the detailed content of some columns. Answer the questions based on the 2 tables:
Read the first table's meta information below regarding 2008 Clasica de San Sebastian to answer the following questions with Python codes.

Meta Information:
Column | Dtype | First Entry | Last Entry
Rank | int64 | 1 | 10
Cyclist | object | Alejandro Valverde (ESP) | David Moncoutie (FRA)
Team | object | Caisse d'Epargne | Cofidis
Time | object | 5h 29' 10 | + 2
UCI ProTour Points | int64 | 40 | 1

Column Details:
Rank | Cyclist | Team | Time | UCI ProTour Points
1 | Alejandro Valverde (ESP) | Caisse d'Epargne | 5h 29' 10 | 40
2 | Alexandr Kolobnev (RUS) | Team CSC Saxo Bank | s.t. | 30
3 | Davide Rebellin (ITA) | Gerolsteiner | s.t. | 25
4 | Paolo Bettini (ITA) | Quick Step | s.t. | 20
5 | Franco Pellizotti (ITA) | Liquigas | s.t. | 15
6 | Denis Menchov (RUS) | Rabobank | s.t. | 11
7 | Samuel Sanchez (ESP) | Euskaltel-Euskadi | s.t. | 7
8 | Stephane Goubert (FRA) | Ag2r-La Mondiale | + 2 | 5
9 | Haimar Zubeldia (ESP) | Euskaltel-Euskadi | + 2 | 3
10 | David Moncoutie (FRA) | Cofidis | + 2 | 1

Question: how many players got less than 10 points?
Answer:
```
#solution in Python
def solution(table_dict):
    import pandas as pd
    df = pd.DataFrame(table_dict)
    #count the number of points less than 10 based on column UCI ProTour Points
    less_than_10_points = df[df["UCI ProTour Points"]< 10].shape[0]
    return less_than_10_points
```
Therefore, the final answer is {4}.

Question: how many cyclists are from Italy?
Answer:
```
#solution in Python
def solution(table_dict):
    import pandas as pd
    df = pd.DataFrame(table_dict)
    #count the number of cyclists who are from Italy
    num_italy_cyclist = df[df["UCI ProTour Points"].str.contains("ITA")].shape[0]
    return num_italy_cyclist
```
Therefore, the final answer is {3}.

Read the second table's meta information below regarding " + <title> + " to to answer the following one question with Python code.
<Meta Information>
<Column Details>
<Question>

Figure 6.3: The task-conducting prompt. The Meta Information and Column Details are highlighted with yellow, and the default answers are highlighted with blue. <title>, <Meta Information>, <Column Details> and <Question> are from the table to be analyzed.

6.3 Experiments and Analysis

6.3.1 Datasets and Experimental Settings

We evaluate our proposed solution on WikiTableQA [36] and TabFact [137] datasets. WikiTableQA and TabFact are two datasets created by Wiki-tables without text context. WikiTableQA mainly contains compositional questions, such as questions requiring counting and ranking table contents. TabFact is a Fact Verification dataset, which can be treated as a special setting of a typical Table-QA problem whose answer set is $\{True, False\}$. Since the proposed solution of this chapter is based on ICL, which is a few-shot learning setting without any training stage, we only use the test set of these two datasets to evaluate the performance, which contains 4344 and 12828 QA pairs, respectively. TabFact dataset further categorizes the test set into simple and complex sets, which include 4219 and 8609 QA pairs, respectively. For the simple set, the QA pairs are usually obtained from a single row or record in the table, reflecting unary facts without complex logical inference. By contrast, for the complex set, the QA pairs are created by information from multiple columns and rows and derived by complex semantic operations, such as argmax, argmin, and the table records are also rewritten to include more semantic understanding. Considering the large size of the TabFact dataset, some studies [37, 38] conducted experiments on a small subset of TabFact, which contains 1,005 simple and 1,019 complex QA pairs. To compare with these studies, we also report the results on this small test subset of TabFact.

6.3.2 Implementation Details and Experimental Results

As many companies and institutions have security concerns of uploading documents to the commercial LLMs, such as GPT-4 [117], we use open-source LLMs, including Mixtral-8x7B [1], Mistral-7B [2], DeepSeek-67B [3] and DeepSeek-7B [4] to conduct our experiments. Since the proposed solution in this chapter is a prompt engineering method, we include Direct Prompting, CoT [31], PoT [30], Binder [37] and Dater [38] as benchmarks, in which Binder and Dater are two solutions leveraging SQL. For the implementation of benchmark methods, we use the implementation of TableCoT[5] [32] for the Direct Prompting and CoT. We re-implemented the PoT method following the example prompts reported

[1] https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1
[2] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2
[3] https://huggingface.co/deepseek-ai/deepseek-llm-67b-chat
[4] https://huggingface.co/deepseek-ai/deepseek-llm-7b-chat
[5] https://github.com/wenhuchen/TableCoT

Table 6.2: Experimental results on WikiTableQA dataset with Exact Match Accuracy as metric.

LLM	Method	EM Acc
Codex	Binder	61.90
	Dater	65.90
Mixtral-8x7B	Direct	53.08
	CoT	53.48
	PoT	40.40
	Tab-PoT	63.33
Mistral-7B	Direct	27.19
	CoT	30.46
	PoT	27.66
	Tab-PoT	52.12
DeepSeek-67B	Direct	54.72
	CoT	55.57
	PoT	43.92
	Tab-PoT	**66.78**
DeepSeek-7B	Direct	33.86
	CoT	34.65
	PoT	19.61
	Tab-PoT	40.03

in PoT [30]. The results of Dater and Binder are directly from study [38]. Besides the benchmarks using LLMs, we also include LogicFactChecker [185], TableFormer [186], OmniTab [104], TAPEX [187] and ReasTAP [188] as benchmark models, all of which follow the pre-training, fine-tuning paradigm. It is worth mentioning that we use greedy decoding for the experiments. At last, even though we employ ICL to provide demonstrations to guide the LLM output of the final results in a "{}," sometimes LLMs can fail to follow this output format. Therefore, we use a simple answer alignment step to post-process the results with incorrect formats. More specifically, we employ a direct prompting method by providing a few demonstrations containing the question, answer and formatted answer following TableCoT [32]. It is worth mentioning that we use the default precision of parameters, namely bfloat16, for the LLMs in this section.

We term our proposed solution as Tab-PoT, and the experimental results are shown in Tables 6.2, 6.3 and 6.4. The experimental results show that our proposed solution can perform competitively compared with state-of-the-art methods, including LLM-based

Table 6.3: Experimental results on TabFact dataset with Exact Match Accuracy as metric. *full* and *small* mean the full and small versions of TabFact dataset.

LLM	Method	Simp_{full}	Comp_{full}	All_{full}	Simp_{small}	Comp_{small}	All_{small}
Codex	Binder	-	-	-	-	-	85.10
	Dater	-	-	-	91.20	80.00	85.60
Mixtral-8x7B	Direct	80.59	69.98	73.47	81.29	70.56	75.89
	CoT	83.53	74.94	77.77	86.07	74.19	80.09
	PoT	73.33	69.07	70.47	76.02	70.66	73.32
	Tab-PoT	86.49	76.06	79.49	88.36	75.07	81.67
Mixtral-7B	Direct	73.57	64.83	67.70	73.13	62.71	67.89
	CoT	73.31	67.52	69.43	73.73	67.12	70.41
	PoT	66.11	63.58	64.41	65.97	66.54	66.25
	Tab-PoT	77.48	66.83	69.75	76.82	66.93	71.84
DeepSeek-67B	Direct	84.36	74.19	77.53	84.68	72.62	78.61
	CoT	87.44	78.00	81.10	88.46	76.84	82.61
	PoT	74.43	71.79	72.65	76.32	72.72	74.51
	Tab-PoT	**90.09**	**78.91**	**82.58**	**91.34**	**80.27**	**85.77**
DeepSeek-7B	Direct	59.16	55.42	56.65	59.50	55.94	57.71
	CoT	69.31	61.18	63.85	70.65	59.76	65.17
	PoT	63.71	58.26	60.06	64.88	58.98	61.91
	Tab-PoT	70.42	62.13	64.86	70.75	61.04	65.86

methods and fine-tuning based models. The Exact Match is employed as the evaluation metric, which is a relatively strict metric because even when the generated answer has the same meaning as the ground truth, it may still be judged as incorrect due to differences in format. The Tab-PoT with DeepSeek-67B can achieve state-of-the-art performance, and the Tab-PoT with both Mixtral-8x7B and DeepSeek-67B can improve the original PoT method by at least 22% on the WikiTableQA dataset. Besides, applying LLMs with a larger number of parameters can significantly improve performance. The CoT does not show many benefits in improving performance compared with direct prompting on the WikiTableQA dataset while significantly improving the performance on the TabFact dataset.

Table 6.4: Comparisons with Fine-tuning based Models using Exact Match Accuracy as metric.

Dataset	Method	EM Acc
WikiTableQA	TableFormer	52.6
	OmniTab	61.2
	TAPEX	57.2
	ReasTAP	58.6
	Tab-Pot	66.8
TabFact_{small}	LogicFactChecker	74.3
	TaPas	83.9
	TAPEX	85.9
	Tab-Pot	85.8

6.3.3 Discussion and Analysis

Ablation Study

As discussed in Section 6.2, our proposed solution consists of three stages: task-planning, task-conducting and task-correction. The task-planning stage can output the relevant columns and reasoning steps to answer the query question, which can be treated as a step of table decomposition and question decomposition, which can also be applied to the conventional PoT. In the task-conducting stage, we prompt the LLM to generate the Python code and a default answer. The default answer is the final answer when the Python code fails to run even after the task-correction stage. Since the default answer is generated after the Python code, it can be treated as an implicit CoT where the Python code is the reasoning rationales in the CoT. Finally, the third stage generates normalization functions to correct the errors in the Python code caused by the heterogeneous data types, which rely on the relevant columns generated by the first stage. Therefore, in this section, we conduct four ablation experiments by applying task-planning, default answer in task-conducting, task-planning and task-correction, and task-planning and default answer in task-conducting to the conventional PoT. The experimental results are shown in Table 6.5, where Plan, Correction and Default represent applying task-planning, task-correction and the default answer in task-conducting stages. We use Mixtral-8x7B as the LLM, and the experimental results demonstrate that each of the proposed three components can significantly improve the PoT baseline.

Table 6.5: Ablation study results on the WikiTableQA dataset with Exact Match Accuracy as metric.

Model	Plan	Correction	Default	EM Acc
PoT				40.40
Ablation 1	✓			46.52
Ablation 2			✓	53.66
Ablation 3	✓	✓		53.31
Ablation 4	✓		✓	58.43
Tab-PoT	✓	✓	✓	63.33

The impact of quantization

Since the LLMs usually have very high hardware requirements for inference, quantization methods are widely used to compact LLMs using lower precision parameters. In this section, we conduct experiments to compare the performance of quantization versions of LLMs. Specifically, similar to the previous section, we also use Mixtral-8x7B to conduct experiments on the WikiTableQA dataset and compare its 16-bit, 8-bit and 4-bit versions of applying the proposed Tab-PoT solution and the experimental results are shown in Table 6.6. For our proposed Tab-Pot, even though applying quantization methods can lead to worse performance, the performance of 8-bit and 4-bit versions is still competitive. It is worth mentioning that the 4-bit version shares a similar RAM footprint with the Mistral-7B model but achieves much higher performance, as shown in Table 6.6 and Table 6.2.

Table 6.6: The impact of LLM quantization methods.

LLM	Parameter Precision	EM Acc
Mixtral-8x7B	16-bit	63.33
	8-bit	62.20
	4-bit	60.52

Analysis on different implementations of PoT

Since Python is a flexible programming language that can implement a function with multiple implementations. In this section, we discuss the differences among these different types of implementations. More specifically, one straightforward implementation is using the Python Standard Library and following the extraction and reasoning steps, as shown

in Figure 4. This method selects relevant data from the table, defines the data with a List of Tuples, and then conducts reasoning over the defined List of Tuples. One obvious drawback of this implementation method is that it needs to repeat the relevant columns in the Python code, which can be very large when the table contains a large number of rows, leading to more inference time. Therefore, a refined method can use the table as the parameter of the solution function, then as shown in Figure 5. At last, since Pandas is a widely used Python library to process tabular data, we can also use Pandas to finish the reasoning tasks with a table dictionary as the input, as shown in Figure 6. We conduct experiments on the WikiTableQA dataset to compare the performance of these three types of implementations. Even though the implementation of applying Python Standard Library can show some benefits regarding the EM Accuracy, it requires more Prompt Tokens and Completion Tokens, as shown in Table 6.7, because this implementation needs to extract relevant from the table directly and define them as a Python dictionary, List or variables. On the other hand, both solutions introducing function parameters can reduce the number of prompting tokens and completion tokens and applying Pandas Library can achieve better performance than using the standard library.

Table 6.7: Ablation study results on the WikiTableQA dataset with Exact Match Accuracy as metric.

Method	EM Acc	#AVG Prompt Tokens	#AVG Completion Tokens
STDLib	44.96	2365	732
STDLib-Para	31.17	2157	114
Pandas	40.40	2241	88

Analysis on inference cost

Since the inference cost is highly correlated with the number of tokens, we use the prompt tokens and generated tokens as the metrics to measure the inference cost in this section. Therefore, we calculate the Average Prompt Tokens and Completion Tokens on the WikiTableQA dataset. As shown in Table 6.8, the proposed Tab-PoT can introduce some overhead compared with other methods regarding the average prompt tokens and average completion tokens on the WikiTableQA dataset. Since the proposed Tab-Pot contains three stages at most, each including instructions and demonstrations, it can reduce the number of Prompt Tokens only when the input table is huge. We group the number of table tokens into 15 bins and plot the relation between the number of table tokens and the prompt tokens on the WikiTableQA dataset. When the number of a table's tokens is larger

than around 1867, our proposed Tab_PoT can use fewer prompt tokens than that of PoT, which means fewer computation operations and less inference time, as shown in Figure 6.4. Since the WikitTableQA dataset contains a large portion of tables whose number of tokens is smaller than 1158, the average prompt tokens of the proposed Tab-PoT is still larger than the one of PoT overall, as shown in Table 6.8. As pointed out by some studies [32], the LLM can perform well on small tables, meaning that we can easily extend the proposed Tab-PoT with other methods, such as CoT, by applying a threshold regarding the size of the input table, to reduce the number of prompt tokens and maintain the performance simultaneously. It is worth mentioning that the price of prompt tokens and completion tokens are different when using commercial LLMs, such as GPT-4 [117].

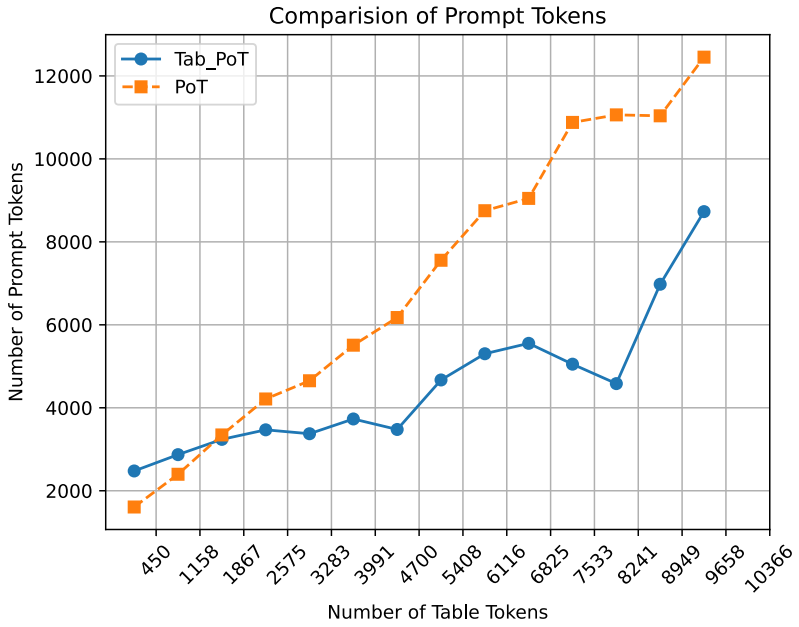


Figure 6.4: Comparison of Prompting Tokens between PoT and Tab_PoT.

6.4 An Example of Deployment

As discussed in the previous sections, the proposed Tab-Pot is a prompting engineering method without further training or fine-tuning. Besides, the experimental results demon-

Table 6.8: Comparisons of Average Prompt Tokens and Completion Tokens.

Method	#AVG Prompt Tokens	#AVG Completion Tokens
Direct	1405	10
CoT	1599	44
PoT	2241	88
Tab-PoT	2022	92

strate that open-source LLMs can also achieve state-of-the-art performance with proper prompting methods. Therefore, the proposed Tab-PoT method can be deployed in a local setting together with other services, which is especially useful when using commercial LLMs facing security concerns. Figure 6.5 shows an example of deployment in the IoT environment. Specifically, a Prompt Management Module and a Post-processing Module are deployed in the edge server, in which the former is responsible for the selection, management and creation of prompts and demonstrations for the requests from IoT devices, and the latter is responsible for post-processing the results generated by the LLMs. With the proposed Prompt Management Module and Post-processing Module, the system can be easily extended to new tasks by adding tailored task-specific prompts in the Task-specific Prompts Database, providing competitive performance for a wide range of tasks.

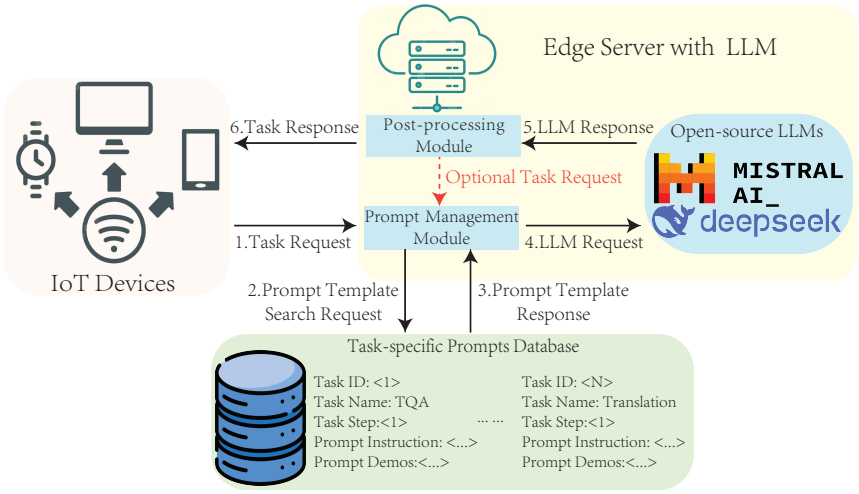


Figure 6.5: An Example of Deployment in the IoT environment.

6.5 Summary of the Chapter

In this chapter, we propose a three-stage prompting solution for the semi-structured Table-QA problem, aiming to alleviate the issues caused by complex structures, heterogeneous data types, huge tables, and the limitations of LLMs. The proposed solution uses a statistics table in the first stage and sub-tables in the following stages, which can reduce the inference cost and improve the performance when the original table is huge. We define a series of atomic operations to guide the demonstration crafting and selection, which can reduce reasoning errors. Besides, we use the task-correction stage to correct the failure code caused by the heterogeneous data types and use a default answer as the final answer when the generated Python code fails to run even after the task-correction step, which can be caused by the complex structure or the limitations of the LLM. Our solution is built on open-source LLMs considering the scenarios with data security considerations. As demonstrated in Section 6.3.2, choosing LLM is crucial. With the number of parameters increasing, the capacities of the LLM can also increase, which also introduces higher hardware requirements. Therefore, instruction tuning for smaller LLMs with tabular data is a future direction to balance the hardware requirements, inference cost and overall performance.

Chapter 7

Conclusion and Future Directions

As the number of visually rich documents surges in business scenarios and on the Internet, the semantic recognition problem on these documents becomes an important research topic to process them automatically and discover valuable knowledge from them. Since tables in visually rich documents and web pages often contain critical information but cannot be processed directly because of the modality gap and their unstructured format, this thesis proposes a complete solution for detecting tables, transforming table images into a text-only structured format and applying prompting method for a semantic recognition task, Table-QA. Since the popular table datasets are only from a limited number of data sources and often contain noisy samples, the models trained with these datasets suffer from their generalization capacities on the data from a new domain. Therefore, this thesis comprehensively discusses the issues of existing datasets for the TD problem, refines the existing datasets, and proposes a new dataset created from the documents from the ICT domain to enrich the data sources. Besides, various benchmarks in different experimental settings, including the cross-domain setting, have been built. The experimental results can demonstrate that the refinement of the existing datasets and the new proposed ICT-TD dataset can improve the generalization ability of models, leading to more robust TD models.

After discussing the TD datasets, a tailored Sparse RCNN-based solution is proposed for the TD problem by improving the proposal initialization, loss functions, and label assignment methods. Since popular object detection models are usually evaluated by the IoU based metrics, such as COCO metrics, the optimization targets of these models are not fully matched with the target of the TD problem because, for the TD problem, a larger bounding box can cover the entire table with less IoU score is preferable than a bounding box with higher IoU score but lost vital information in the table. Therefore, this thesis

introduces the Information Coverage Score (ICS) to evaluate the detection performance and proposes an ICS-based loss function to guide the model in avoiding information loss during training. Besides, many other characteristics of the TD problem are considered to improve the Sparse R-CNN based model, such as the sparse distribution of tables in the document images and initializing the region proposals with image sizes. The experimental results demonstrate that the proposed ICS score and the proposed ICS-based loss function can lead to better table extraction results, and the tailored detection model can achieve state-of-the-art performance on various datasets.

For the TSR task of transforming table images into text-only structured or semi-structured formats, this thesis revisits the limitations of current detection-based solutions. First, the problem formulations of many studies are not enough to fully recover the complex structures of tables. Second, the characteristics of detection models are not fully considered in the problem formulation and model training. Therefore, this thesis identifies critical designs for the success of the TSR problem, including proper problem formulation, building long-dependency features, and addressing the multi-label classification formulation. Based on the observations, a tailored Cascade R-CNN based TSR model is proposed with adjusted region proposals, a new multi-label classification formulation, and modules to build long-dependency features, achieving state-of-the-art performance.

At last, a training-free prompting method for the Table-QA problem is proposed as a demonstration to solve the semantic recognition task, considering the inference cost, the limitations of LLMs, and the challenges caused by the complex table structures, heterogeneous data types, huge tables, and limitations of LLMs. The proposed prompting method utilizes Python to extend the LLM and divide the Table-QA problem into three tasks, including task-planning, task-conducting and task-correction stages, which can avoid including huge tables as a part of the prompt and correct the generated Python code which fails to run. Comprehensive experiments are conducted for all the models proposed in this thesis, and the experimental results can demonstrate that the proposed models can outperform state-of-the-art models by a large margin. Besides, each designed component in the proposed solutions is carefully analyzed to reveal their effectiveness and efficiency.

Even though the proposed methods for the TD, TSR, and Table-QA can show significant benefits compared with benchmark models, there are still some limitations to the models proposed in this thesis. First, the proposed models for the TD and TSR tasks are designed for well-formatted tables, meaning their performance might be degraded when the target table has an irregular shape, such as distorted tables. Second, as the proposed pipeline consists of multiple steps, it has the issue of error accumulation, meaning that the performance of the subsequent model in this pipeline depends on the models preceding it. Third, because of the longer pipeline and the selected detection models, the inference time

is also larger than some benchmark models, such as one-stage detection models. Finally, for the Table-QA task, we only considered the Table-QA problem in a single table setting, which means that our problem formulation cannot deal with cases requiring information from multiple input tables.

Considering these limitations, models in each step can be further improved in many directions. For the TD datasets, more datasets from other data sources can be further added following the data creation protocols proposed in this thesis. For the TD model, the proposed method follows the training of typical object models in the supervised manager, while unsupervised methods, such as Self-Supervised Learning (SSL) [189], can be further integrated to use large-scale documents without labels. For example, a backbone network can be trained by SSL with a large scale of unlabeled documents, and a typical detection model with the backbone can be fine-tuned with labelled datasets. Besides, domain adaptation can also be a direction to be explored to mitigate the issues caused by the domain variance for the TD problem. For the TSR task, the model proposed in this thesis is detection-based, while other types of models, such as transformer-based encoder-decoder models, can also achieve promising performance. Encoder-decoder models are suitable for integrating OCR capacities, making a fully end-to-end solution possible for the TSR task. And SSL pre-training methods can also be beneficial for the TSR models. Another promising direction is applying graph-based TSR models to deal with the tables with irregular shapes. For the Table-QA problem, a training-free prompting method, which contains three prompting stages, is proposed. Even though the proposed method can reduce the training cost, its performance heavily relies on the capacities of the base LLM, and it increases the inference cost because of its larger number of prompt tokens. Therefore, the fine-tuning of the LLM with fewer number parameters can be a direction when enough computation resources are available. The performance of the Table-QA can also be further improved by LLM-based Agent. For example, the provided table and its question can be the Agent's environment, and information extraction, reasoning and arithmetic calculations can be defined as actions. Then, the Agent can use the demonstrations in the Memory module to solve the questions by leveraging the defined actions based on the environment and use a Reflection module to further refine the prompts. Besides these directions for improving the models in each step, there are other important topics in document understanding. For example, figures are also widely used to summarize critical information in various documents, making the problem of recognizing and understanding figures in documents also necessary. Understanding the long text information from documents is also challenging and essential because of the limitations of current LLMs, where Retrieval-Augmented Generation (RAG) can be a solution. Lastly, the solution reported in this thesis is to transform tables in images into text format and utilize text-only LLMs. Multi-modal LLMs have

also attracted much research attention recently, and they can feed the table images into the multi-modal LLM directly to obtain the final results. Some multi-modal models have demonstrated promising potential for visually rich document understanding tasks, which is also a good direction to explore further.

References

[1] Max Göbel, Tamir Hassan, Ermelinda Oro, and Giorgio Orsi. Icdar 2013 table competition. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1449–1453. IEEE, 2013.

[2] Liangcai Gao, Xiaohan Yi, Zhuoren Jiang, Leipeng Hao, and Zhi Tang. Icdar2017 competition on page object detection. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 1417–1422. IEEE, 2017.

[3] Liangcai Gao, Yilun Huang, Hervé Déjean, Jean-Luc Meunier, Qinqin Yan, Yu Fang, Florian Kleber, and Eva Lang. Icdar 2019 competition on table detection and recognition (ctdar). In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1510–1515. IEEE, 2019.

[4] Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou, and Zhoujun Li. Tablebank: Table benchmark for image-based table detection and recognition. In *Proceedings of The 12th language resources and evaluation conference*, pages 1918–1925, 2020.

[5] Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. Publaynet: largest dataset ever for document layout analysis. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1015–1022. IEEE, 2019.

[6] Jing Fang, Xin Tao, Zhi Tang, Ruiheng Qiu, and Ying Liu. Dataset, ground-truth and performance metrics for table detection evaluation. In *2012 10th IAPR International Workshop on Document Analysis Systems*, pages 445–449. IEEE, 2012.

[7] Abdelrahman Abdallah, Alexander Berendeyev, Islam Nuradin, and Daniyar Nurseitov. Tncr: Table net detection and classification dataset. *Neurocomputing*, 473:79–97, 2022.

[8] Johan Fernandes, Murat Simsek, Burak Kantarci, and Shahzad Khan. Tabledet: An end-to-end deep learning approach for table detection and table image classification in data sheet images. *Neurocomputing*, 468:317–334, 2022.

[9] Devashish Prasad, Ayan Gadpal, Kshitij Kapadni, Manish Visave, and Kavita Sultanpure. Cascadetabnet: An approach for end to end table detection and structure recognition from image-based documents. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 572–573, 2020.

[10] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14454–14463, 2021.

[11] Zheng Ge, Songtao Liu, Zeming Li, Osamu Yoshie, and Jian Sun. Ota: Optimal transport assignment for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 303–312, 2021.

[12] Qinghang Hong, Fengming Liu, Dong Li, Ji Liu, Lu Tian, and Yi Shan. Dynamic sparse r-cnn. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4723–4732, 2022.

[13] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo. Diffusiondet: Diffusion model for object detection. *arXiv preprint arXiv:2211.09788*, 2022.

[14] Yuming Du, Wen Guo, Yang Xiao, and Vincent Lepetit. 1st place solution for the uvo challenge on image-based open-world segmentation 2021. *arXiv preprint arXiv:2110.10239*, 2021.

[15] Zheng Ge, Songtao Liu, Feng Wang, Zeming Li, and Jian Sun. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*, 2021.

[16] Nam Tuan Ly and Atsuhiro Takasu. An end-to-end multi-task learning model for image-based table recognition. pages 626–634, 2023.

[17] Jiaquan Ye, Xianbiao Qi, Yelin He, Yihao Chen, Dengyi Gu, Peng Gao, and Rong Xiao. Pingan-vcgroup's solution for icdar 2021 competition on scientific literature parsing task b: Table recognition to html. *arXiv preprint arXiv:2105.01848*, 2021.

[18] Zhanghui Kuang, Hongbin Sun, Zhizhong Li, Xiaoyu Yue, Tsui Hin Lin, Jianyong Chen, Huaqiang Wei, Yiqin Zhu, Tong Gao, Wenwei Zhang, Kai Chen, Wayne Zhang,

and Dahua Lin. Mmocr: A comprehensive toolbox for text detection, recognition and understanding. *arXiv preprint arXiv:2108.06543*, 2021.

[19] JaidedAI. Easyocr. https://github.com/JaidedAI/EasyOCR.git, 2022.

[20] Huawen Shen, Xiang Gao, Jin Wei, Liang Qiao, Yu Zhou, Qiang Li, and Zhanzhan Cheng. Divide rows and conquer cells: Towards structure recognition for large tables. IJCAI, 2023.

[21] Xu Zhong, Elaheh ShafieiBavani, and Antonio Jimeno Yepes. Image-based table recognition: data, model, and evaluation. In *European conference on computer vision*, pages 564–580. Springer, 2020.

[22] Shoaib Ahmed Siddiqui, Imran Ali Fateh, Syed Tahseen Raza Rizvi, Andreas Dengel, and Sheraz Ahmed. Deeptabstr: Deep learning based table structure recognition. In *2019 international conference on document analysis and recognition (ICDAR)*, pages 1403–1409. IEEE, 2019.

[23] Khurram Azeem Hashmi, Didier Stricker, Marcus Liwicki, Muhammad Noman Afzal, and Muhammad Zeshan Afzal. Guided table structure recognition through anchor optimization. *IEEE Access*, 9:113521–113534, 2021.

[24] Brandon Smock, Rohith Pesala, and Robin Abraham. Pubtables-1m: Towards comprehensive table extraction from unstructured documents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4634–4642, 2022.

[25] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6154–6162, 2018.

[26] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[27] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017.

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects

in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[29] Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. Pal: Program-aided language models. In *International Conference on Machine Learning*, pages 10764–10799. PMLR, 2023.

[30] Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*, 2023.

[31] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

[32] Wenhu Chen. Large language models are few (1)-shot table reasoners. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1120–1130, 2023.

[33] The pandas development team. pandas-dev/pandas: Pandas, February 2020.

[34] Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61, 2010.

[35] Yilun Zhao, Yunxiang Li, Chenying Li, and Rui Zhang. MultiHiertt: Numerical reasoning over multi hierarchical tabular and textual data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6588–6600, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[36] Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480, 2015.

[37] Zhoujun Cheng, Tianbao Xie, Peng Shi, Chengzu Li, Rahul Nadkarni, Yushi Hu, Caiming Xiong, Dragomir Radev, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. Binding language models in symbolic languages. In *The Eleventh International Conference on Learning Representations*, 2023.

[38] Yunhu Ye, Binyuan Hui, Min Yang, Binhua Li, Fei Huang, and Yongbin Li. Large language models are versatile decomposers: Decomposing evidence and questions for table-based reasoning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 174–184, 2023.

[39] Tianze Shi, Chen Zhao, Jordan Boyd-Graber, Hal Daumé III, and Lillian Lee. On the potential of lexico-logical alignments for semantic parsing to sql queries. *arXiv preprint arXiv:2010.11246*, 2020.

[40] Yunjia Zhang, Jordan Henkel, Avrilia Floratou, Joyce Cahoon, Shaleen Deep, and Jignesh M Patel. Reactable: Enhancing react for table question answering. *arXiv preprint arXiv:2310.00815*, 2023.

[41] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

[42] Fangyu Lei, Tongxu Luo, Pengqi Yang, Weihao Liu, Hanwen Liu, Jiahe Lei, Yiming Huang, Yifan Wei, Shizhu He, Jun Zhao, et al. Tableqakit: A comprehensive and practical toolkit for table-based question answering. *arXiv preprint arXiv:2310.15075*, 2023.

[43] Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. Table meets llm: Can large language models understand structured table data? a benchmark and empirical study. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 645–654, 2024.

[44] Naihao Deng, Zhenjie Sun, Ruiqi He, Aman Sikka, Yulong Chen, Lin Ma, Yue Zhang, and Rada Mihalcea. Tables as images? exploring the strengths and limitations of llms on multimodal representations of tabular data. *arXiv preprint arXiv:2402.12424*, 2024.

[45] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot arena: An open platform for evaluating llms by human preference, 2024.

[46] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph,

Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. 2021.

[47] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[48] Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. Complexity-based prompting for multi-step reasoning. *arXiv preprint arXiv:2210.00720*, 2022.

[49] Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. Towards understanding chain-of-thought prompting: An empirical study of what matters. In *The 61st Annual Meeting of the Association for Computational Linguistics*, 2023.

[50] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[51] Mohammad Javad Shafiee, Brendan Chywl, Francis Li, and Alexander Wong. Fast yolo: A fast you only look once system for real-time embedded object detection in video. *arXiv preprint arXiv:1709.05943*, 2017.

[52] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[53] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[54] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[55] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019.

[56] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[57] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[58] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.

[59] Shilong Zhang, Xinjiang Wang, Jiaqi Wang, Jiangmiao Pang, Chengqi Lyu, Wenwei Zhang, Ping Luo, and Kai Chen. Dense distinct query for end-to-end object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7329–7338. IEEE, 2023.

[60] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[61] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

[62] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. Dab-detr: Dynamic anchor boxes are better queries for detr. *arXiv preprint arXiv:2201.12329*, 2022.

[63] Peize Sun, Yi Jiang, Enze Xie, Wenqi Shao, Zehuan Yuan, Changhu Wang, and Ping Luo. What makes for end-to-end object detection? In *International Conference on Machine Learning*, pages 9934–9944. PMLR, 2021.

[64] Fangyi Chen, Han Zhang, Kai Hu, Yu-Kai Huang, Chenchen Zhu, and Marios Savvides. Enhanced training of query-based object detection via selective query recollection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23756–23765. IEEE, 2023.

[65] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[66] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666, 2019.

[67] Tu Zheng, Shuai Zhao, Yang Liu, Zili Liu, and Deng Cai. Scaloss: Side and corner aligned loss for bounding box regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3535–3543, 2022.

[68] Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas Huang. Unitbox: An advanced object detection network. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 516–520, 2016.

[69] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-iou loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 12993–13000, 2020.

[70] Zhaohui Zheng, Ping Wang, Dongwei Ren, Wei Liu, Rongguang Ye, Qinghua Hu, and Wangmeng Zuo. Enhancing geometric factors in model learning and inference for object detection and instance segmentation. *IEEE Transactions on Cybernetics*, 52(8):8574–8586, 2021.

[71] Yihui He, Chenchen Zhu, Jianren Wang, Marios Savvides, and Xiangyu Zhang. Bounding box regression with uncertainty for accurate object detection. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 2888–2897, 2019.

[72] Ajoy Mondal, Peter Lipps, and CV Jawahar. Iiit-ar-13k: A new dataset for graphical object detection in documents. In *Document Analysis Systems: 14th IAPR International Workshop, DAS 2020, Wuhan, China, July 26–29, 2020, Proceedings 14*, pages 216–230. Springer, 2020.

[73] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: high quality object detection and instance segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 43(5):1483–1498, 2019.

[74] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3349–3364, 2020.

[75] Yilun Huang, Qinqin Yan, Yibo Li, Yifan Chen, Xiong Wang, Liangcai Gao, and Zhi Tang. A yolo-based table detection method. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 813–818. IEEE, 2019.

[76] Sebastian Schreiber, Stefan Agne, Ivo Wolf, Andreas Dengel, and Sheraz Ahmed. Deepdesrt: Deep learning for detection and structure recognition of tables in document images. In *Intl. Conf. on document analysis and recognition*, volume 1, pages 1162–1167. IEEE, 2017.

[77] Shoaib Ahmed Siddiqui, Muhammad Imran Malik, Stefan Agne, Andreas Dengel, and Sheraz Ahmed. Decnt: Deep deformable cnn for table detection. *IEEE access*, 6:74151–74161, 2018.

[78] Ertugrul Kara, Mark Traquair, Murat Simsek, Burak Kantarci, and Shahzad Khan. Holistic design for deep learning-based discovery of tabular structures in datasheet images. *Engineering Applications of Artificial Intelligence*, 90:103551, 2020.

[79] Zewen Chi, Heyan Huang, Heng-Da Xu, Houjin Yu, Wanxuan Yin, and Xian-Ling Mao. Complicated table structure recognition. *arXiv preprint arXiv:1908.04729*, 2019.

[80] Darshan Adiga, Shabir Ahmad Bhat, Muzaffar Bashir Shah, and Viveka Vyeth. Table structure recognition based on cell relationship, a bottom-up approach. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 1–8. INCOMA Ltd., 2019.

[81] Xinyi Zheng, Douglas Burdick, Lucian Popa, Xu Zhong, and Nancy Xin Ru Wang. Global table extractor (gte): A framework for joint table identification and cell structure recognition using visual context. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 697–706, 2021.

[82] Sebastian Schreiber, Stefan Agne, Ivo Wolf, Andreas R. Dengel, and Sheraz Ahmed. Deepdesrt: Deep learning for detection and structure recognition of tables in document images. *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 01:1162–1167, 2017.

[83] Hao Liu, Xin Li, Bing Liu, Deqiang Jiang, Yinsong Liu, and Bo Ren. Neural collaborative graph machines for table structure recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4533–4542. IEEE, 2022.

[84] Ning Lu, Wenwen Yu, Xianbiao Qi, Yihao Chen, Ping Gong, Rong Xiao, and Xiang Bai. MASTER: Multi-aspect non-local network for scene text recognition. *Pattern Recognition*, 2021.

[85] Yongshuai Huang, Ning Lu, Dapeng Chen, Yibo Li, Zecheng Xie, Shenggao Zhu, Liangcai Gao, and Wei Peng. Improving table structure recognition with visual-alignment sequential coordinate modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11134–11143, 2023.

[86] Johan Fernandes, Bin Xiao, Murat Simsek, Burak Kantarci, Shahzad Khan, and Ala Abu Alkheir. Tablestrrec: framework for table structure recognition in data sheet images. *International Journal on Document Analysis and Recognition (IJDAR)*, pages 1–19, 2023.

[87] Wenyuan Xue, Baosheng Yu, Wen Wang, Dacheng Tao, and Qingyong Li. Tgrnet: A table graph reconstruction network for table structure recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1295–1304. IEEE, 2021.

[88] Xiao-Hui Li, Fei Yin, He-Sen Dai, and Cheng-Lin Liu. Table structure recognition and form parsing by end-to-end object detection and relation parsing. *Pattern Recognition*, 132:108946, 2022.

[89] Liang Qiao, Zaisheng Li, Zhanzhan Cheng, Peng Zhang, Shiliang Pu, Yi Niu, Wenqi Ren, Wenming Tan, and Fei Wu. Lgpma: Complicated table structure recognition with local and global pyramid mask alignment. In *International conference on document analysis and recognition*, pages 99–114. Springer, 2021.

[90] Chris Tensmeyer, Vlad I Morariu, Brian Price, Scott Cohen, and Tony Martinez. Deep splitting and merging for table structure decomposition. In *Intl Conf on Document Analysis and Recognition*, pages 114–121. IEEE, 2019.

[91] Zhenrong Zhang, Jianshu Zhang, Jun Du, and Fengren Wang. Split, embed and merge: An accurate table structure recognizer. *Pattern Recognition*, 126:108565, 2022.

[92] Nam Quan Nguyen, Anh Duy Le, Anh Khoa Lu, Xuan Toan Mai, and Tuan Anh Tran. Formerge: Recover spanning cells in complex table structure using transformer network. In *International Conference on Document Analysis and Recognition*, pages 522–534. Springer, 2023.

[93] Jiawei Wang, Weihong Lin, Chixiang Ma, Mingze Li, Zheng Sun, Lei Sun, and Qiang Huo. Robust table structure recognition with dynamic queries enhanced detection transformer. *Pattern Recognition*, 144:109817, 2023.

[94] Chixiang Ma, Weihong Lin, Lei Sun, and Qiang Huo. Robust table detection and structure recognition from heterogeneous document images. *Pattern Recognition*, 133:109006, 2023.

[95] Jacob Devlin et al. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

[96] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[97] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2019.

[98] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[99] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, 2020.

[100] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. Ernie: Enhanced language representation with informative entities. *arXiv preprint arXiv:1905.07129*, 2019.

[101] Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*, 2019.

[102] Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Wang. Hybridqa: A dataset of multi-hop question answering over tabular and textual data. *arXiv preprint arXiv:2004.07347*, 2020.

[103] Jonathan Herzig, Paweł Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Martin Eisenschlos. Tapas: Weakly supervised table parsing via pre-training. *arXiv preprint arXiv:2004.02349*, 2020.

[104] Zhengbao Jiang, Yi Mao, Pengcheng He, Graham Neubig, and Weizhu Chen. Omnitab: Pretraining with natural and synthetic data for few-shot table-based question answering. *arXiv preprint arXiv:2207.03637*, 2022.

[105] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1192–1200, 2020.

[106] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, et al. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. *arXiv preprint arXiv:2012.14740*, 2020.

[107] Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. Ocr-free document understanding transformer. In *European Conference on Computer Vision (ECCV)*, 2022.

[108] Qiming Peng, Yinxu Pan, Wenjin Wang, Bin Luo, Zhenyu Zhang, Zhengjie Huang, Teng Hu, Weichong Yin, Yongfeng Chen, Yin Zhang, et al. Ernie-layout: Layout knowledge enhanced pre-training for visually-rich document understanding. *arXiv preprint arXiv:2210.06155*, 2022.

[109] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[110] Chatgpt. https://chat.openai.com/. Accessed: 12 07, 2023.

[111] Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, et al. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*, 2023.

[112] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

[113] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

[114] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023.

[115] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[116] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Annual Meeting of the Association for Computational Linguistics*, 2022.

[117] OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023.

[118] Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 9:1, 2023.

[119] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023.

[120] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

[121] Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*, 2023.

[122] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

[123] Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. Monkey: Image resolution and text label are important things for large multi-modal models. *arXiv preprint arXiv:2311.06607*, 2023.

[124] Ruyi Xu, Yuan Yao, Zonghao Guo, Junbo Cui, Zanlin Ni, Chunjiang Ge, Tat-Seng Chua, Zhiyuan Liu, and Gao Huang. LLaVA-UHD: an lmm perceiving any aspect ratio and high-resolution images. *arXiv preprint arXiv:2403.11703*, 2024.

[125] Tianyu Yu, Haoye Zhang, Yuan Yao, Yunkai Dang, Da Chen, Xiaoman Lu, Ganqu Cui, Taiwen He, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. Rlaif-v: Aligning mllms through open-source ai feedback for super gpt-4v trustworthiness. *arXiv preprint arXiv:2405.17220*, 2024.

[126] Tianyu Yu, Yuan Yao, Haoye Zhang, Taiwen He, Yifeng Han, Ganqu Cui, Jinyi Hu, Zhiyuan Liu, Hai-Tao Zheng, Maosong Sun, et al. Rlhf-v: Towards trustworthy mllms via behavior alignment from fine-grained correctional human feedback. *arXiv preprint arXiv:2312.00849*, 2023.

[127] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[128] Yukang Chen, Shengju Qian, Haotian Tang, Xin Lai, Zhijian Liu, Song Han, and Jiaya Jia. Longlora: Efficient fine-tuning of long-context large language models. *arXiv preprint arXiv:2309.12307*, 2023.

[129] Tianshu Zhang, Xiang Yue, Yifei Li, and Huan Sun. Tablellama: Towards open large generalist models for tables. *arXiv preprint arXiv:2311.09206*, 2023.

[130] Fengbin Zhu, Ziyang Liu, Fuli Feng, Chao Wang, Moxin Li, and Tat-Seng Chua. Tat-llm: A specialized language model for discrete reasoning over tabular and textual data. *arXiv preprint arXiv:2401.13223*, 2024.

[131] Weizhe Lin, Rexhina Blloshmi, Bill Byrne, Adrià de Gispert, and Gonzalo Iglesias. An inner table retriever for robust table question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9909–9926, 2023.

[132] Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Wayne Xin Zhao, and Ji rong Wen. Structgpt: A general framework for large language model to reason over structured data. In *Conference on Empirical Methods in Natural Language Processing*, 2023.

[133] Pragya Srivastava, Manuj Malik, and Tanuja Ganu. Assessing llms' mathematical reasoning in financial document question answering. *arXiv preprint arXiv:2402.11194*, 2024.

[134] Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. Search-based neural structured learning for sequential question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1821–1831, 2017.

[135] Linyong Nan, Chiachun Hsieh, Ziming Mao, Xi Victoria Lin, Neha Verma, Rui Zhang, Wojciech Kryściński, Hailey Schoelkopf, Riley Kong, Xiangru Tang, et al. Fetaqa: Free-form table question answering. *Transactions of the Association for Computational Linguistics*, 10:35–49, 2022.

[136] Nancy XR Wang, Diwakar Mahajan, Marina Danilevsky, and Sara Rosenthal. Semeval-2021 task 9: Fact verification and evidence finding for tabular data in scientific documents (sem-tab-facts). *arXiv preprint arXiv:2105.13995*, 2021.

[137] Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. Tabfact: A large-scale dataset for table-based fact verification. *arXiv preprint arXiv:1909.02164*, 2019.

[138] Rami Aly, Zhijiang Guo, Michael Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. Feverous: Fact

extraction and verification over unstructured and structured information. *arXiv preprint arXiv:2106.05707*, 2021.

[139] Xiao Li, Yawei Sun, and Gong Cheng. Tsqa: tabular scenario based question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13297–13305, 2021.

[140] Alon Talmor, Ori Yoran, Amnon Catav, Dan Lahav, Yizhong Wang, Akari Asai, Gabriel Ilharco, Hannaneh Hajishirzi, and Jonathan Berant. Multimodalqa: Complex question answering over text, tables and images. *arXiv preprint arXiv:2104.06039*, 2021.

[141] Sachin Raja, Ajoy Mondal, and CV Jawahar. Icdar 2023 competition on visual question answering on business document images. In *International Conference on Document Analysis and Recognition*, pages 454–470. Springer, 2023.

[142] Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations (ICLR 2023)*, 2023.

[143] Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. Synthetic prompting: Generating chain-of-thought demonstrations for large language models. *arXiv preprint arXiv:2302.00618*, 2023.

[144] Kashun Shum, Shizhe Diao, and Tong Zhang. Automatic prompt augmentation and selection with chain-of-thought from labeled data. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12113–12139, Singapore, December 2023. Association for Computational Linguistics.

[145] Ori Yoran, Tomer Wolfson, Ben Bogin, Uri Katz, Daniel Deutch, and Jonathan Berant. Answering questions by meta-reasoning over multiple chains of thought. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5942–5966, Singapore, December 2023. Association for Computational Linguistics.

[146] Yihan Cao, Shuyi Chen, Ryan Liu, Zhiruo Wang, and Daniel Fried. Api-assisted code generation for question answering on varied table structures. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14536–14548, 2023.

[147] Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 68539–68551. Curran Associates, Inc., 2023.

[148] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[149] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[150] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

[151] detrex contributors. detrex: An research platform for transformer-based object detection algorithms. https://github.com/IDEA-Research/detrex, 2022.

[152] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[153] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[154] Tianbo Lu, Xiaobo Guo, Bing Xu, Lingling Zhao, Yong Peng, and Hongyu Yang. Next big thing in big data: The security of the ict supply chain. In *2013 International Conference on Social Computing*, pages 1066–1073, 2013.

[155] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[156] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569, 2017.

[157] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[158] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. You only learn one representation: Unified network for multiple tasks. *arXiv preprint arXiv:2105.04206*, 2021.

[159] Glenn Jocher. Ultralytics yolov5, 2020.

[160] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*, 2022.

[161] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolov8, 2023.

[162] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[163] Devashish Prasad, Ayan Gadpal, Kshitij Kapadni, Manish Visave, and Kavita Sultanpure. Cascadetabnet: An approach for end to end table detection and structure recognition from image-based documents, 2020.

[164] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[165] Ajoy Mondal, Madhav Agarwal, and CV Jawahar. Dataset agnostic document object detection. *Pattern Recognition*, 142:109698, 2023.

[166] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11963–11975, 2022.

[167] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *arXiv preprint arXiv:2201.03545*, 2022.

[168] Meng-Hao Guo, Cheng-Ze Lu, Qibin Hou, Zhengning Liu, Ming-Ming Cheng, and Shi-Min Hu. Segnext: Rethinking convolutional attention design for semantic segmentation. *Advances in Neural Information Processing Systems*, 35:1140–1156, 2022.

[169] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[170] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[171] Brandon Smock, Rohith Pesala, and Robin Abraham. Aligning benchmark datasets for table structure recognition. *arXiv preprint arXiv:2303.00716*, 2023.

[172] Brandon Smock and Rohith Pesala. Table Transformer, 06 2021.

[173] Ahmed Nassar, Nikolaos Livathinos, Maksym Lysak, and Peter Staar. Tableformer: Table structure understanding with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4614–4623, 2022.

[174] Tianhe Ren, Shilong Liu, Feng Li, Hao Zhang, Ailing Zeng, Jie Yang, Xingyu Liao, Ding Jia, Hongyang Li, He Cao, Jianan Wang, Zhaoyang Zeng, Xianbiao Qi, Yuhui Yuan, Jianwei Yang, and Lei Zhang. detrex: Benchmarking detection transformers, 2023.

[175] Thanapon Noraset, Lalita Lowphansirikul, and Suppawong Tuarob. Wabiqa: A wikipedia-based thai question-answering system. *Information processing & management*, 58(1):102431, 2021.

[176] Zhiyi Luo, Yingying Zhang, and Shuyun Luo. A token-based transition-aware joint framework for multi-span question answering. *Information Processing & Management*, 61(3):103678, 2024.

[177] Dieu-Hien Nguyen, Nguyen-Khang Le, and Le-Minh Nguyen. Viwiqa: Efficient end-to-end vietnamese wikipedia-based open-domain question-answering systems for single-hop and multi-hop questions. *Information Processing & Management*, 60(6):103514, 2023.

[178] Xing Cao, Yun Liu, and Feng Sun. Predict, pretrained, select and answer: Interpretable and scalable complex question answering over knowledge bases. *Knowledge-Based Systems*, 278:110820, 2023.

[179] Jinhao Zhang, Lizong Zhang, Bei Hui, and Ling Tian. Improving complex knowledge base question answering via structural information learning. *Knowledge-Based Systems*, 242:108252, 2022.

[180] Guangyou Zhou, Zhiwen Xie, Zongfu Yu, and Jimmy Xiangji Huang. Dfm: A parameter-shared deep fused model for knowledge base question answering. *Information Sciences*, 547:103–118, 2021.

[181] Ali Vosoughi, Shijian Deng, Songyang Zhang, Yapeng Tian, Chenliang Xu, and Jiebo Luo. Cross modality bias in visual question answering: A causal view with possible worlds vqa. *IEEE Transactions on Multimedia*, 2024.

[182] Fuwei Zhang, Ruomei Wang, Fan Zhou, Yuanmao Luo, and Jinyu Li. Psam: Parameter-free spatiotemporal attention mechanism for video question answering. *IEEE Transactions on Multimedia*, 2023.

[183] Zihan Yu, Liang He, Zhen Wu, Xinyu Dai, and Jiajun Chen. Towards better chain-of-thought prompting strategies: A survey. *arXiv preprint arXiv:2310.04959*, 2023.

[184] Bowen Zhao, Changkai Ji, Yuejie Zhang, Wen He, Yingwen Wang, Qing Wang, Rui Feng, and Xiaobo Zhang. Large language models are complex table parsers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14786–14802, 2023.

[185] Wanjun Zhong, Duyu Tang, Zhangyin Feng, Nan Duan, Ming Zhou, Ming Gong, Linjun Shou, Daxin Jiang, Jiahai Wang, and Jian Yin. Logicalfactchecker: Leveraging logical operations for fact checking with graph module network. *arXiv preprint arXiv:2004.13659*, 2020.

[186] Jingfeng Yang, Aditya Gupta, Shyam Upadhyay, Luheng He, Rahul Goel, and Shachi Paul. Tableformer: Robust transformer modeling for table-text encoding. *arXiv preprint arXiv:2203.00274*, 2022.

[187] Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. Tapex: Table pre-training via learning a neural sql executor. *arXiv preprint arXiv:2107.07653*, 2021.

[188] Yilun Zhao, Linyong Nan, Zhenting Qi, Rui Zhang, and Dragomir Radev. Reastap: Injecting table reasoning skills during pre-training via synthetic reasoning examples. *arXiv preprint arXiv:2210.12374*, 2022.

[189] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[190] Shubham Singh Paliwal, D Vishwanath, Rohit Rahul, Monika Sharma, and Lovekesh Vig. Tablenet: Deep learning model for end-to-end table detection and tabular data extraction from scanned document images. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 128–133. IEEE, 2019.

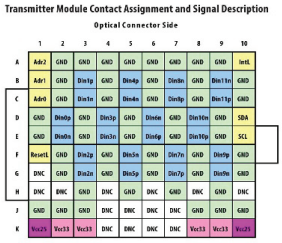
[191] Yibo Li, Liangcai Gao, Zhi Tang, Qinqin Yan, and Yilun Huang. A gan-based feature generator for table detection. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 763–768. IEEE, 2019.

APPENDICES

.1 Appendix for Revisiting Table Detection Datasets (Chapter 3)

.1.1 Visualization result

This appendix section is to visualize and provide information on the ground truth of the test sample in Figure 3.8, as shown in Figure 1.



Transmitter Module Contact Assignment and Signal Description

Optical Connector Side

	1	2	3	4	5	6	7	8	9	10
A	Adr2	GND	GND	GND	GND	GND	GND	GND	GND	IntL
B	Adr1	GND	Din1p	GND	Din4p	GND	Din8n	GND	Din11n	GND
C	Adr0	GND	Din1n	GND	Din4n	GND	Din8p	GND	Din11p	GND
D	GND	Din0p	GND	Din3p	GND	Din6n	GND	Din10n	GND	SDA
E	GND	Din0n	GND	Din3n	GND	Din6p	GND	Din10p	GND	SCL
F	ResetL	GND	Din2p	GND	Din5n	GND	Din7n	GND	Din9p	GND
G	DNC	GND	Din2n	GND	Din5p	GND	Din7p	GND	Din9n	GND
H	DNC	DNC	GND	DNC	GND	DNC	GND	DNC	GND	DNC
J	GND	GND	GND	DNC	DNC	DNC	DNC	GND	GND	GND
K	Vcc25	Vcc33	Vcc33	DNC	DNC	DNC	DNC	Vcc33	Vcc33	Vcc25

Figure 9. Host Board Pattern for Transmitter Connector – Top View

Signal Name	Signal Description	I/O	Type
Adr[2:0]	TWS Module Bus Address bits: Address has the form 0101hjkx where Adr2, Adr1 & Adr0 correspond to h, j & k respectively and x corresponds to the R/W command.	I	3.3V LVTTL
Din[11:0]p	Transmitter Data Non-inverting Input for channels 11 through 0	I	CML
Din[11:0]n	Transmitter Data Inverting Input for channels 11 through 0	I	CML
DNC	Reserved – Do Not Connect to any electrical potential on Host PCB		
GND	Signal Common: All module voltages are referenced to this potential unless otherwise stated. Connect these pins directly to the host board signal ground plane.		
IntL	Interrupt signal to Host. Asserted Low: An interrupt is generated in response to any Tx Fault condition, loss of input signal or assertion of any monitor Flag. It may be programmed through the TWS interface to generate either a pulse or static level with pulse mode as default. This output presents a High-Z condition when IntL is de-asserted and requires a pull-up on the Host board. Pull-up to the Host 3.3 V supply is recommended.	O	3.3V LVTTL, high-Z or driven to 0 level
ResetL	Reset signal to module, Asserted Low: When asserted the optical outputs are disabled, TWS interface commands are inhibited, and the module returns to default and non-volatile settings. An internal pullup biases the input High if the input is open.	I	3.3V LVTTL
SDA	TWS interface data signal: Pull-up with a 2.0 kΩ to 8.0 kΩ resistor to the Host 3.3 V supply is recommended.	I/O	3.3V LVTTL/ Open-Drain
SCL	TWS interface clock signal I: Pull-up with a 2.0 kΩ to 8.0 kΩ resistor to the Host 3.3 V supply is recommended.	I	3.3V LVTTL
Vcc25	2.5V Power supply, External common connection of pins required – not common internally		
Vcc33	3.3 V Power supply, External common connection of pins required – not common internally		
Case Common	Not accessible in connector. Case common incorporates exposed conductive surfaces including threaded bosses and is electrically isolated from signal common, i.e. GND. Connect as appropriate for EMI shield integrity. See EMI clip and bezel cutout recommendation.		

Figure 1: Ground truth of the testing sample in Figure 3.8.

.2 Appendix for Table Detection (Chapter 4)

.2.1 Model implementations and settings

In this section, we list the implementations and configuration files of the baseline models, including RetinaNet [157], FCOS [55], YOLOX-X [15], YOLOR-X [158], YOLOv5-X [159], YOLOv7-X [160], YOLOv8-X [161], FasterR-CNN [56], DiffusionDet [13], Deformable-DETR [61], and SparseR-CNN [10], as summarized in Table 1. It is worth mentioning that we modified the training epochs of the listed configuration files, trained FasterR-CNN, MaskR-CNN, DiffusionDet, Deformable-DETR, and SparseR-CNN for 120 epochs, and trained other one-stage detectors for 300 epochs.

Table 1: Summary of model implementations and settings

Model	Implementation	Setting File
RetinaNet	Detectron2	https://github.com/facebookresearch/detectron2/blob/main/configs/COCO-Detection/retinanet_R_50_FPN_3x.yaml
FCOS	Detectron2	https://github.com/facebookresearch/detectron2/blob/main/configs/COCO-Detection/fcos_R_50_FPN_1x.py
YOLOX	Official codebase	https://github.com/Megvii-BaseDetection/YOLOX/blob/main/exps/default/yolox_x.py
YOLOR	Official codebase	https://github.com/WongKinYiu/yolor/blob/main/cfg/yolor_csp_x.cfg
YOLOv5	Official codebase	https://github.com/ultralytics/ultralytics/blob/main/ultralytics/cfg/models/v5/yolov5.yaml
YOLOv7	Official codebase	https://github.com/WongKinYiu/yolov7/blob/main/cfg/training/yolov7x.yaml
YOLOv8	Official codebase	https://github.com/ultralytics/ultralytics/blob/main/ultralytics/cfg/models/v8/yolov8.yaml

FasterR-CNN	Detectron2	https://github.com/facebookresearch/detectron2/blob/main/configs/COCO-Detection/faster_rcnn_R_50_FPN_3x.yaml
MaskR-CNN	Detectron2	https://github.com/facebookresearch/detectron2/blob/main/configs/COCO-InstanceSegmentation/mask_rcnn_R_50_FPN_3x.yaml
DiffusionDet	Official codebase	https://github.com/ShoufaChen/DiffusionDet/blob/main/configs/diffdet.coco.res50.300boxes.yaml
Deformable-DETR	detrex	https://github.com/IDEA-Research/detrex/blob/main/projects/deformable_detr/configs/deformable_detr_r50_two_stage_50ep.py
SparseR-CNN	Official codebase	https://github.com/PeizeSun/SparseR-CNN/blob/main/projects/SparseRCNN/configs/sparsercnn.res50.300pro.3x.yaml

.2.2 Compared with other Table Detection models

In this section, we include the experimental results using the evaluation protocols in ICDAR2013, ICDAR2017, and ICT-TD datasets. More specifically, for the ICDAR2013 dataset, the F1 score thresholded by 50% is the competition evaluation metric. For the ICDAR2017 dataset, Precision, Recall, and F1 scores thresholded by 60% and 80% are used as evaluation metrics. For the ICT-TD dataset, Weighted Average F1 score, as defined in Equation 3.5, is used as the evaluation metric whose thresholds are 80%, 85%, 90%, and 95%. The experimental results of them are shown in Table 2, 3 and 3.4. It is worth mentioning that the experimental results of TableDet [8], DeCNT [77], YOLOv3-TD [75], DeepDeSRT [76], TableNet [190], GAN-TD [191], in Table 2, 3 are from study [8].

Table 2: Experimental results on ICDAR2013 dataset (IoU = 50%).

Model	Precision	Recall	F1
CascadeTabNet [9]	100	100	100
TableDet [8]	100	100	100
DeCNT [77]	99.6	99.6	99.6
YOLOv3-TD [75]	94.9	100	97.3
DeepDeSRT [76]	97.4	96.2	96.8
TableNet [190]	97.0	96.3	96.6
SparseTableDet (Proposed)	100	100	100

Table 3: Experimental results on ICDAR2017 dataset.

IoU Threshold	Model	Precision	Recall	F1
60%	TableDet [8]	98.8	99.7	99.3
	YOLOv3-TD [75]	97.2	97.8	97.5
	DeCNT [77]	96.5	97.1	96.8
	GAN-TD [191]	94.4	94.4	94.4
	SparseTableDet (Proposed)	**99.1**	**100.0**	**99.5**
80%	TableDet [8]	**97.4**	98.4	97.9
	YOLOv3-TD [75]	96.8	97.5	97.1
	DeCNT [77]	96.7	93.7	95.2
	GAN-TD [191]	90.3	90.3	90.3
	SparseTableDet (Proposed)	96.7	**99.7**	**98.2**

.2.3 Detailed experimental results

In this appendix section, we include the detailed experimental results on the TNCR and ICT-TD datasets, as shown in Table 7 and Table 8. Besides, we also include some prediction results for the models trained with IoU-based and ICS-based losses, as discussed in section 4.3.2.

Table 4: Experimental results on the ICT-TD dataset.

Model	F1				WAvg. F1
	IoU(80%)	IoU(85%)	IoU(90%)	IoU(95%)	
TableDet	93.6	91.6	89.1	75.7	87.1
DiffusionDet	95.5	94.2	91.1	76.4	88.9
Deformable-DETR	95.0	93.9	91.2	**83.0**	90.5
SparseR-CNN	94.3	93.0	90.4	78.8	88.8
SparseTableDet	**97.2**	**96.4**	**94.2**	81.8	**92.1**

Table 5: Detailed Experimental results on the ICDAR2017 dataset.

Method		IoU										
		50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	Avg.
RetinaNet	P.	96.4	96.4	95.2	94.9	92.1	90.7	90.3	88.1	85.5	75.0	90.4
	R.	97.8	97.8	96.9	96.3	94.7	93.8	93.2	91.6	89.1	80.1	93.1
	F1	97.1	97.1	96.0	95.6	93.4	92.2	91.7	89.8	87.3	77.5	91.7
FCOS	P.	97.3	96.3	95.2	94.9	92.4	90.8	90.2	87.6	84.9	72.7	90.2
	R.	98.1	97.5	96.9	96.6	95.3	94.7	94.1	92.8	90.3	82.6	93.9
	F1	97.7	96.9	96.0	95.7	93.8	92.7	92.1	90.1	87.5	77.3	92.0
YOLOX-X	P.	97.0	97.0	96.3	95.5	94.1	93.3	91.2	87.1	78.8	54.2	88.5
	R.	99.4	99.4	99.1	98.4	96.9	95.6	93.5	89.4	81.3	59.5	91.3
	F1	98.2	98.2	97.7	96.9	95.5	94.4	92.3	88.2	80.0	56.7	89.9
YOLOR-X	P.	97.5	97.4	96.4	95.1	94.3	93.4	92.2	92.2	88.1	79.9	92.6
	R.	98.8	98.8	97.8	96.9	96.3	95.3	94.7	94.4	91.3	84.1	94.8
	F1	98.1	98.1	97.1	96.0	95.3	94.3	93.4	93.3	89.7	81.9	93.7
YOLOV5-X	P.	98.3	98.3	97.9	97.9	95.8	95.4	94.2	93.8	90.2	82.8	94.4
	R.	99.7	99.7	99.1	99.1	97.8	97.2	96.6	96.3	93.8	86.6	96.6
	F1	99.0	99.0	98.5	98.5	96.8	96.3	95.4	95.0	91.9	84.7	95.5
YOLOV7-X	P.	98.0	97.1	97.0	95.9	95.3	94.1	93.3	93.3	89.8	80.7	93.5
	R.	99.1	98.4	98.1	97.5	97.2	96.6	96.0	95.6	93.2	85.1	95.7
	F1	98.5	97.7	97.5	96.7	96.2	95.3	94.6	94.4	91.5	82.8	94.6
YOLOV8-X	P.	99.0	97.9	97.1	96.5	95.2	94.6	94.1	93.9	91.0	80.6	94.0
	R.	100.0	99.1	98.8	98.1	97.2	96.9	96.6	96.3	94.1	85.1	96.2
	F1	99.5	98.5	97.9	97.3	96.2	95.7	95.3	95.1	92.5	82.8	95.1

Continued on next page

Table 5 Detailed Experimental results on the ICDAR2017 dataset (continued from previous page).

Method		IoU										
		50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	Avg.
FasterR-CNN	P.	97.8	96.9	96.7	96.6	95.4	94.0	92.9	91.3	88.2	79.4	92.9
	R.	98.1	97.8	97.5	97.2	96.6	95.3	94.7	93.8	91.0	83.2	94.5
	F1	97.9	97.3	97.1	96.9	96.0	94.6	93.8	92.5	89.6	81.3	93.7
MaskR-CNN	P.	97.5	96.5	96.2	96.2	95.1	94.0	93.9	92.6	89.8	68.5	92.0
	R.	98.1	97.8	97.5	97.2	96.9	96.0	95.6	94.7	92.5	76.6	94.3
	F1	97.8	97.1	96.8	96.7	96.0	95.0	94.7	93.6	91.1	72.3	93.1
TableDet	P.	99.4	98.6	98.5	98.5	96.4	95.1	94.1	93.5	89.0	78.8	94.2
	R.	100.0	99.7	99.1	99.1	97.8	96.6	96.0	95.0	91.9	84.1	95.9
	F1	99.7	99.1	98.8	98.8	97.1	95.8	95.0	94.2	90.4	81.4	95.0
DiffusionDet	P.	98.1	98.0	97.6	97.1	96.1	94.6	93.9	92.5	89.1	77.8	93.5
	R.	99.7	99.7	99.1	98.8	97.8	96.3	96.0	94.4	91.6	83.5	95.7
	F1	98.9	98.8	98.3	97.9	96.9	95.4	94.9	93.4	90.3	80.5	94.6
Deformable-DETR	P.	97.2	97.2	96.8	96.8	95.8	93.7	92.6	91.3	89.7	78.6	93.0
	R.	98.8	98.4	98.1	98.1	97.5	96.6	96.3	95.0	93.2	83.2	95.5
	F1	98.0	97.8	97.4	97.4	96.6	95.1	94.4	93.1	87.1	80.8	94.2
SparseR-CNN	P.	99.3	98.3	97.8	97.8	97.0	95.9	94.8	94.8	92.7	84.6	95.3
	R.	100.0	99.7	99.1	99.1	98.8	98.1	97.5	97.2	95.3	89.1	97.4
	F1	99.6	99.0	98.4	98.4	97.9	97.0	96.1	96.0	94.0	86.8	96.3
SparseTableDet (Proposed)	P.	99.7	99.7	99.1	99.1	98.9	97.5	96.7	95.6	93.1	83.6	96.3
	R.	100.0	100.0	100.0	100.0	100.0	100.0	99.7	99.1	96.6	89.1	98.4
	F1	99.8	99.8	99.5	99.5	99.4	98.7	98.2	97.3	94.8	86.3	97.3

Table 6: Detailed Experimental results on the ICDAR2019 dataset.

Method		IoU										
		50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	Avg.
RetinaNet	P.	98.6	98.4	97.4	97.2	96.2	94.3	93.9	90.5	85.3	72.8	92.5
	R.	99.6	99.1	98.7	98.2	97.3	96.0	95.1	92.7	88.4	78.6	94.4
	F1	99.1	98.7	98.0	97.7	96.7	95.1	94.5	91.6	86.8	75.6	93.4
FCOS	P.	97.1	96.7	96.7	95.7	95.5	94.4	92.2	90.2	83.1	63.7	90.5
	R.	98.9	98.4	98.4	97.8	97.6	96.9	95.1	93.5	88.4	75.7	94.1

Continued on next page

Table 6 Detailed Experimental results on the ICDAR2019 dataset (continued from previous page).

Method		IoU										
		50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	Avg.
	F1	98.0	97.5	97.5	96.7	96.5	95.6	93.6	91.8	85.7	69.2	92.3
YOLOX-X	P.	97.5	97.1	96.3	95.7	95.3	94.3	94.1	92.2	88.3	70.4	92.1
	R.	98.7	98.4	98.0	97.1	96.7	96.0	95.1	93.3	90.2	74.4	93.8
	F1	98.1	97.7	97.1	96.4	96.0	95.1	94.6	92.7	89.2	72.3	92.9
YOLOR-X	P.	98.7	98.7	98.2	98.2	97.5	97.5	96.6	95.6	92.5	82.1	95.5
	R.	99.6	99.6	99.1	99.1	98.9	98.7	97.8	96.4	94.2	85.3	96.9
	F1	99.1	99.1	98.6	98.6	98.2	98.1	97.2	96.0	93.3	83.7	96.2
YOLOV5-X	P.	98.7	98.6	98.5	98.5	98.5	98.4	97.5	96.5	95.1	83.7	96.4
	R.	99.8	99.8	99.6	99.3	99.3	99.3	98.9	97.8	96.4	86.6	97.7
	F1	99.2	99.2	99.0	98.9	98.9	98.8	98.2	97.1	95.7	85.1	97.0
YOLOV7-X	P.	99.5	98.6	98.6	98.4	98.1	97.7	97.6	96.7	93.3	81.0	95.9
	R.	100.0	99.8	99.8	99.1	99.1	98.7	98.4	97.8	94.9	84.9	97.2
	F1	99.7	99.2	99.2	98.7	98.6	98.2	98.0	97.2	94.1	82.9	96.5
YOLOV8-X	P.	99.0	99.0	98.8	98.8	98.8	98.7	97.7	96.8	94.0	89.5	97.1
	R.	99.8	99.8	99.6	99.3	99.3	99.1	98.4	98.0	95.3	92.0	98.1
	F1	99.4	99.4	99.2	99.0	99.0	98.9	98.0	97.4	94.6	90.7	97.6
FasterR-CNN	P.	97.9	97.8	96.8	96.8	95.6	94.5	94.5	93.5	89.7	76.0	93.3
	R.	98.7	98.4	98.0	97.6	96.9	95.8	95.6	94.7	91.1	80.2	94.7
	F1	98.3	98.1	97.4	97.2	96.2	95.1	95.0	94.1	90.4	78.0	94.0
MaskR-CNN	P.	98.9	97.8	97.7	96.5	96.4	95.4	95.4	94.5	91.2	71.2	93.5
	R.	99.3	98.9	98.7	98.0	97.6	96.9	96.2	95.8	92.7	76.6	95.1
	F1	99.1	98.3	98.2	97.2	97.0	96.1	95.8	95.1	91.9	73.8	94.3
TableDet	P.	98.5	97.5	97.5	97.4	96.3	95.3	94.4	93.5	90.7	77.2	93.8
	R.	99.1	98.9	98.7	98.4	97.3	96.4	95.3	94.4	92.2	83.7	95.5
	F1	98.8	98.2	98.1	97.9	96.8	95.8	94.8	93.9	91.4	80.3	94.6
DiffusionDet	P.	98.5	98.3	98.3	97.6	96.4	96.0	95.0	92.6	90.3	74.4	93.7
	R.	99.8	99.6	99.6	99.3	98.4	97.6	96.7	94.4	92.4	82.2	96.0
	F1	99.1	98.9	98.9	98.4	97.4	96.8	95.8	93.5	91.3	78.1	94.8
Deformable-DETR	P.	98.7	97.9	97.6	97.1	96.9	96.5	95.3	94.1	91.0	80.2	94.5
	R.	99.6	99.3	99.1	98.9	98.9	98.7	97.8	96.7	94.4	86.6	97.0
	F1	99.1	98.6	98.3	98.0	97.9	97.6	96.5	95.4	92.7	83.3	95.7
SparseR-CNN	P.	98.4	97.9	97.9	97.9	97.1	96.5	96.4	96.1	93.5	86.0	95.8

Continued on next page

Table 6 Detailed Experimental results on the ICDAR2019 dataset (continued from previous page).

Method		IoU										
		50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	Avg.
	R.	99.8	99.6	99.3	99.3	99.1	98.9	98.7	98.2	96.2	91.8	98.1
	F1	99.1	98.7	98.6	98.6	98.1	97.7	97.5	97.1	94.8	88.8	96.9
SparseTableDet	P.	99.5	98.8	98.8	98.8	98.6	98.5	98.3	97.5	95.2	88.6	97.3
(Proposed)	R.	100.0	99.8	99.6	99.6	99.6	99.6	99.6	98.9	97.3	92.0	98.6
	F1	99.7	99.3	99.2	99.2	99.1	99.0	98.9	98.2	96.3	90.3	97.9

Table 7: Detailed Experimental results on the TNCR dataset.

Method		IoU										
		50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	Avg.
RetinaNet	P.	89.7	89.7	89.6	89.4	88.9	88.6	87.5	85.6	81.5	69.8	86.0
	R.	96.2	96.2	96.1	96.0	95.3	95.0	94.1	92.3	88.2	78.2	92.8
	F1	92.8	92.8	92.7	92.6	92.0	91.7	90.6	88.8	84.8	73.8	89.3
FCOS	P.	87.8	87.7	87.6	87.3	86.7	86.3	85.6	83.1	79.4	68.5	84.0
	R.	94.4	94.3	94.2	94.0	93.4	93.1	92.3	90.4	87.5	78.4	91.2
	F1	91.0	90.9	90.8	90.5	89.9	89.6	88.8	86.6	83.3	73.1	87.5
YOLOX-X	P.	87.1	86.8	86.5	86.1	85.5	84.4	83.0	80.6	76.7	57.9	81.5
	R.	96.0	95.6	95.0	94.2	93.4	91.8	89.5	86.7	82.7	65.9	89.1
	F1	91.3	91.0	90.6	90.0	89.3	87.9	86.1	83.5	79.6	61.6	85.1
YOLOR-X	P.	90.4	90.3	90.2	90.1	89.6	89.2	88.2	86.0	82.9	75.1	87.2
	R.	98.7	98.6	98.5	98.3	97.6	97.1	95.6	93.3	90.1	83.6	95.1
	F1	94.4	94.3	94.2	94.0	93.4	93.0	91.8	89.5	86.4	79.1	91.0
YOLOV5-X	P.	93.0	92.8	92.7	92.4	92.2	91.8	91.1	88.9	86.0	79.3	90.0
	R.	99.3	99.2	99.1	99.0	98.9	98.5	97.9	96.1	93.4	88.2	97.0
	F1	96.0	95.9	95.8	95.6	95.4	95.0	94.4	92.4	89.5	83.5	93.4
YOLOV7-X	P.	92.0	91.9	91.8	91.7	91.6	91.3	90.3	88.6	85.8	77.3	89.2
	R.	99.1	99.0	98.9	98.9	98.8	98.4	97.3	96.0	93.1	86.4	96.6
	F1	95.4	95.3	95.2	95.2	95.1	94.7	93.7	92.2	89.3	81.6	92.8
YOLOV8-X	P.	93.1	93.1	93.0	92.8	92.4	92.2	91.4	89.5	86.6	79.2	90.3
	R.	99.3	99.3	99.3	99.2	98.9	98.6	98.0	96.3	93.9	87.6	97.0
	F1	96.1	96.1	96.0	95.9	95.5	95.3	94.6	92.8	90.1	83.2	93.5

Continued on next page

Table 7 Detailed Experimental results on the TNCR dataset (continued from previous page).

Method		IoU										
		50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	Avg.
FasterR-CNN	P.	89.1	89.1	88.9	88.7	88.5	88.2	87.8	86.5	81.5	66.6	85.5
	R.	94.3	94.3	94.2	94.1	93.7	93.4	93.0	91.7	87.5	76.1	91.2
	F1	91.6	91.6	91.5	91.3	91.0	90.7	90.3	89.0	84.4	71.0	88.3
MaskR-CNN	P.	90.2	90.0	90.0	89.8	89.7	89.1	88.2	86.3	81.7	64.4	85.9
	R.	95.3	95.2	95.1	95.0	94.8	94.3	93.7	92.0	88.0	75.5	91.9
	F1	92.7	92.5	92.5	92.3	92.2	91.6	90.9	89.1	84.7	69.5	88.8
TableDet	P.	91.7	91.7	91.7	91.5	91.3	90.7	90.2	88.4	84.0	72.9	88.4
	R.	98.1	98.1	98.0	97.9	97.7	97.0	96.6	95.2	91.7	83.6	95.4
	F1	94.8	94.8	94.7	94.6	94.4	93.7	93.3	91.7	87.7	77.9	91.8
Deformable-DETR	P.	90.3	90.3	90.1	90.1	89.8	89.2	88.5	86.8	84.4	77.2	87.7
	R.	99.2	99.1	99.0	98.9	98.8	98.6	97.9	97.0	94.7	89.4	97.3
	F1	94.5	94.5	94.3	94.3	94.1	93.7	93.0	91.6	89.3	82.9	92.3
DiffusionDet	P.	91.7	91.6	91.6	91.3	90.8	90.2	89.4	87.7	84.6	74.0	88.3
	R.	99.6	99.6	99.6	99.4	98.7	97.9	97.2	95.5	92.9	85.2	96.6
	F1	95.5	95.4	95.4	95.2	94.6	93.9	93.1	91.4	88.5	79.2	92.3
SparseR-CNN	P.	90.9	90.9	90.7	90.7	90.5	90.3	89.8	89.0	85.5	77.1	88.6
	R.	99.9	99.8	99.8	99.7	99.7	99.7	99.5	98.7	97.1	89.9	98.4
	F1	95.2	95.1	95.0	95.0	94.9	94.8	94.4	93.6	90.9	83.0	93.2
SparseTableDet (Proposed)	P.	93.0	93.0	92.9	92.9	92.7	92.5	92.2	91.3	88.8	77.2	90.6
	R.	100.0	100.0	100.0	99.9	99.9	99.8	99.6	99.0	96.9	87.9	98.3
	F1	96.4	96.4	96.3	96.3	96.2	96.0	95.8	95.0	92.7	82.2	94.3

Table 8: Detailed Experimental results on the ICT-TD dataset.

Method		IoU										
		50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	Avg.
RetinaNet	P.	95.8	95.7	94.7	94.4	92.4	91.1	90.0	87.9	82.1	68.0	89.2
	R.	97.3	97.2	96.9	96.2	94.8	93.3	92.1	90.7	85.4	72.1	91.6
	F1	96.5	96.4	95.8	95.3	93.6	92.2	91.0	89.3	83.7	70.0	90.4
FCOS	P.	92.0	91.8	90.8	90.5	89.4	88.1	86.8	84.4	80.5	66.8	86.1
	R.	93.6	93.1	92.8	92.1	91.5	90.6	89.0	87.4	84.1	73.7	88.8

Continued on next page

Table 8 Detailed Experimental results on the ICT-TD dataset (continued from previous page).

Method		IoU										
		50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	Avg.
	F1	92.8	92.4	91.8	91.3	90.4	89.3	87.9	85.9	82.3	70.1	87.4
YOLOX-X	P.	95.8	94.9	94.4	93.6	92.2	90.7	88.9	86.2	80.4	64.5	88.2
	R.	98.7	97.8	97.3	96.7	95.1	93.5	91.4	88.3	82.9	67.9	91.0
	F1	97.2	96.3	95.8	95.1	93.6	92.1	90.1	87.2	81.6	66.2	89.6
YOLOR-X	P.	97.6	97.3	96.4	95.5	95.0	94.2	93.3	90.7	88.3	79.1	92.7
	R.	99.4	99.1	98.6	97.9	97.1	96.5	95.3	92.1	89.8	81.1	94.7
	F1	98.5	98.2	97.5	96.7	96.0	95.3	94.3	91.4	89.0	80.1	93.7
YOLOV5-X	P.	97.4	97.2	97.2	97.0	96.4	95.6	94.8	93.7	90.7	81.2	94.1
	R.	98.9	98.8	98.8	98.6	98.0	97.4	96.8	95.8	92.8	83.8	96.0
	F1	98.1	98.0	98.0	97.8	97.2	96.5	95.8	94.7	91.7	82.5	95.0
YOLOV7-X	P.	98.2	98.2	98.0	97.9	96.8	95.8	94.8	93.7	91.8	80.9	94.6
	R.	99.5	99.4	99.3	99.1	98.5	97.8	96.7	95.6	93.4	83.5	96.3
	F1	98.8	98.8	98.6	98.5	97.6	96.8	95.7	94.6	92.6	82.2	95.4
YOLOV8-X	P.	97.9	97.1	97.0	96.9	96.4	95.6	94.7	93.6	91.4	82.4	94.3
	R.	99.1	98.8	98.7	98.6	98.1	97.5	96.6	95.7	93.2	84.9	96.1
	F1	98.5	97.9	97.8	97.7	97.2	96.5	95.6	94.6	92.3	83.6	95.2
FasterR-CNN	P.	96.6	96.5	96.5	95.4	94.2	93.2	92.1	90.0	86.0	73.6	91.4
	R.	97.4	97.2	97.1	96.4	95.3	94.8	93.7	91.7	87.6	76.5	92.8
	F1	97.0	96.8	96.8	95.9	94.7	94.0	92.9	90.8	86.8	75.0	92.1
MaskR-CNN	P.	96.6	96.5	95.5	95.3	94.2	93.1	92.1	90.0	86.9	74.4	91.5
	R.	97.4	97.1	96.9	96.3	95.5	94.4	93.4	91.7	88.9	77.8	92.9
	F1	97.0	96.8	96.2	95.8	94.8	93.7	92.7	90.8	87.9	76.1	92.2
TableDet	P.	97.4	96.4	96.3	96.3	95.1	94.0	92.9	90.5	88.2	72.5	92.0
	R.	98.2	97.9	97.5	97.2	96.3	95.5	94.4	92.7	90.1	79.3	93.9
	F1	97.8	97.1	96.9	96.7	95.7	94.7	93.6	91.6	89.1	75.7	92.9
DiffusionDet	P.	96.5	96.4	96.3	95.8	95.2	94.5	93.9	92.5	89.2	73.6	92.4
	R.	99.3	99.2	99.0	98.8	98.4	97.8	97.2	96.0	93.1	79.5	95.8
	F1	97.9	97.8	97.6	97.3	96.8	96.1	95.5	94.2	91.1	76.4	94.1
Deformable-DETR	P.	97.0	96.6	96.3	96.0	95.2	94.4	93.7	92.6	89.7	80.3	93.2
	R.	98.9	98.6	98.5	98.3	97.8	96.9	96.3	95.2	92.8	85.8	95.9
	F1	97.9	97.6	97.4	97.1	96.5	95.6	95.0	93.9	91.2	83.0	94.5
SparseR-CNN	P.	96.2	96.0	95.5	95.3	94.2	93.3	92.6	91.1	88.3	75.6	91.8

Continued on next page

Table 8 Detailed Experimental results on the ICT-TD dataset (continued from previous page).

Method		IoU										
		50%	55%	60%	65%	70%	75%	80%	85%	90%	95%	Avg.
	R.	99.0	98.9	98.7	98.4	97.7	97.0	96.2	94.9	92.5	82.4	95.6
	F1	97.6	97.4	97.1	96.8	95.9	95.1	94.3	93.0	90.4	78.8	93.7
SparseTableDet	P.	97.5	97.4	97.1	96.9	96.7	96.2	96.0	95.1	92.8	79.6	94.5
(Proposed)	R.	99.3	99.3	99.3	99.2	99.2	98.7	98.5	97.8	95.6	84.1	97.1
	F1	98.4	98.3	98.2	98.0	97.9	97.4	97.2	96.4	94.2	81.8	95.8

.3 Model Prediction Visualization

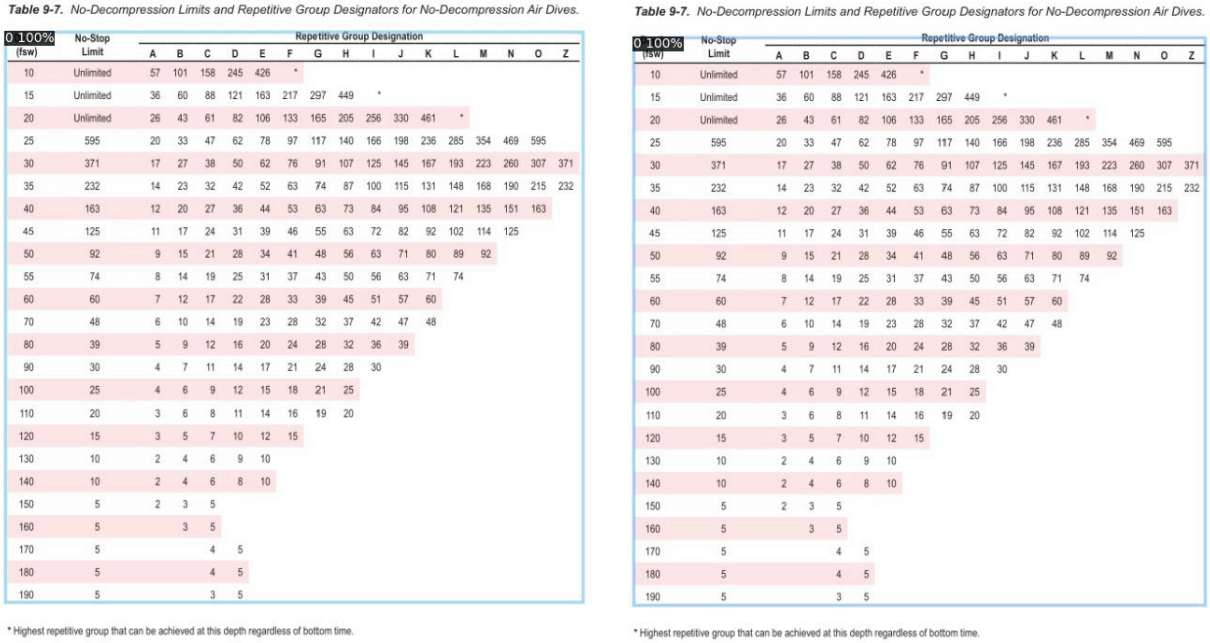


Table 9-7. No-Decompression Limits and Repetitive Group Designators for No-Decompression Air Dives.

0 100% (fsw)	No-Stop Limit	Repetitive Group Designation															
		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	Z
10	Unlimited	57	101	158	245	426	*										
15	Unlimited	36	60	88	121	163	217	297	449	*							
20	Unlimited	26	43	61	82	106	133	165	205	256	330	461	*				
25	595	20	33	47	62	78	97	117	140	166	198	236	285	354	469	595	
30	371	17	27	38	50	62	76	91	107	125	145	167	193	223	260	307	371
35	232	14	23	32	42	52	63	74	87	100	115	131	148	168	190	215	232
40	163	12	20	27	36	44	53	63	73	84	95	108	121	135	151	163	
45	125	11	17	24	31	39	46	55	63	72	82	92	102	114	125		
50	92	9	15	21	28	34	41	48	56	63	71	80	89	92			
55	74	8	14	19	25	31	37	43	50	56	63	71	74				
60	60	7	12	17	22	28	33	39	45	51	57	60					
70	48	6	10	14	19	23	28	32	37	42	47	48					
80	39	5	9	12	16	20	24	28	32	36	39						
90	30	4	7	11	14	17	21	24	28	30							
100	25	4	6	9	12	15	18	21	25								
110	20	3	6	8	11	14	16	19	20								
120	15	3	5	7	10	12	15										
130	10	2	4	6	9	10											
140	10	2	4	6	8	10											
150	5	2	3	5													
160	5		3	5													
170	5			4	5												
180	5			4	5												
190	5			3	5												

* Highest repetitive group that can be achieved at this depth regardless of bottom time.

(a) ICS-based loss

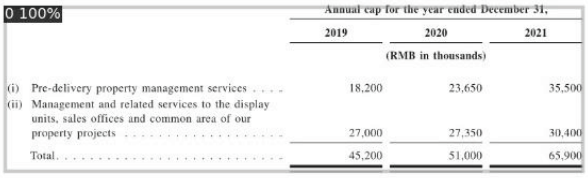
Table 9-7. No-Decompression Limits and Repetitive Group Designators for No-Decompression Air Dives.

0 100% (fsw)	No-Stop Limit	Repetitive Group Designation															
		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	Z
10	Unlimited	57	101	158	245	426	*										
15	Unlimited	36	60	88	121	163	217	297	449	*							
20	Unlimited	26	43	61	82	106	133	165	205	256	330	461	*				
25	595	20	33	47	62	78	97	117	140	166	198	236	285	354	469	595	
30	371	17	27	38	50	62	76	91	107	125	145	167	193	223	260	307	371
35	232	14	23	32	42	52	63	74	87	100	115	131	148	168	190	215	232
40	163	12	20	27	36	44	53	63	73	84	95	108	121	135	151	163	
45	125	11	17	24	31	39	46	55	63	72	82	92	102	114	125		
50	92	9	15	21	28	34	41	48	56	63	71	80	89	92			
55	74	8	14	19	25	31	37	43	50	56	63	71	74				
60	60	7	12	17	22	28	33	39	45	51	57	60					
70	48	6	10	14	19	23	28	32	37	42	47	48					
80	39	5	9	12	16	20	24	28	32	36	39						
90	30	4	7	11	14	17	21	24	28	30							
100	25	4	6	9	12	15	18	21	25								
110	20	3	6	8	11	14	16	19	20								
120	15	3	5	7	10	12	15										
130	10	2	4	6	9	10											
140	10	2	4	6	8	10											
150	5	2	3	5													
160	5		3	5													
170	5			4	5												
180	5			4	5												
190	5			3	5												

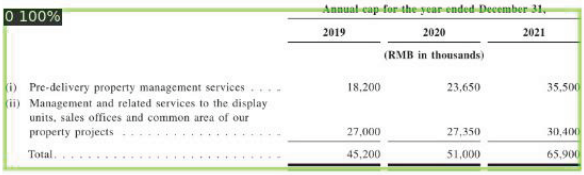
* Highest repetitive group that can be achieved at this depth regardless of bottom time.

(b) IoU-based loss

Figure 2: Prediction samples of models trained with ICS-based loss and IoU-based loss.



(a) ICS-based loss



(b) IoU-based loss

Figure 3: Prediction samples of models trained with ICS-based loss and IoU-based loss.

.4 Appendix for Table Question Answering (Chapter 6)

```
Answer the questions based on the two given tables:

Rank | Cyclist              | Team            | Time     | UCI ProTour Points
1    | Alejandro Valverde (ESP) | Caisse d'Epargne  | 5h 29' 10 | 40
2    | Alexandr Kolobnev (RUS)  | Team CSC Saxo Bank | s.t.     | 30
3    | Davide Rebellin (ITA)    | Gerolsteiner      | s.t.     | 25
4    | Paolo Bettini (ITA)      | Quick Step        | s.t.     | 20
5    | Franco Pellizotti (ITA)  | Liquigas          | s.t.     | 15
6    | Denis Menchov (RUS)      | Rabobank          | s.t.     | 11
7    | Samuel Sanchez (ESP)     | Euskaltel-Euskadi | s.t.     | 7
8    | Stephane Goubert (FRA)   | Ag2r-La Mondiale  | + 2      | 5
9    | Haimar Zubeldia (ESP)    | Euskaltel-Euskadi | + 2      | 3
10   | David Moncoutie (FRA)    | Cofidis           | + 2      | 1

Question: who is the cyclist after Alexandr Kolobnev in the table?
Answer:
  def solution():
    # Define the table as a list of cyclist names
    table = [
      "Alejandro Valverde (ESP)",
      "Alexandr Kolobnev (RUS)",
      "Davide Rebellin (ITA)",
      "Paolo Bettini (ITA)",
      "Franco Pellizotti (ITA)",
      "Denis Menchov (RUS)",
      "Samuel Sanchez (ESP)",
      "Stephane Goubert (FRA)",
      "Haimar Zubeldia (ESP)",
      "David Moncoutie (FRA)"]
    # Find the cyclist after Alexandr Kolobnev
    kolobnev_index = table.index("Alexandr Kolobnev (RUS)")
    if kolobnev_index != -1 and kolobnev_index < len(table) - 1:
      next_cyclist = table[kolobnev_index + 1]
      return next_cyclist
    else:
      return "Alexandr Kolobnev is either not in the table or is the last entry."
  Therefore, the final answer with python code is {print(solution())}
Question: which country had the most cyclists finish within the top 10?
Answer:
  def solution():
    # Define the table as a list of tuples containing cyclist names and their countries
    table = [
      ("Alejandro Valverde (ESP)", "ESP"),
      ("Alexandr Kolobnev (RUS)", "RUS"),
      ("Davide Rebellin (ITA)", "ITA"),
      ("Paolo Bettini (ITA)", "ITA"),
      ("Franco Pellizotti (ITA)", "ITA"),
      ("Denis Menchov (RUS)", "RUS"),
      ("Samuel Sanchez (ESP)", "ESP"),
      ("Stephane Goubert (FRA)", "FRA"),
      ("Haimar Zubeldia (ESP)", "ESP"),
      ("David Moncoutie (FRA)", "FRA")]
    # Count the occurrences of each country in the top 10
    country_count = {}
    for _, country in table:
      if country in country_count:
        country_count[country] += 1
      else:
        country_count[country] = 1
    # Find the country with the most cyclists in the top 10
    max_country = max(country_count, key=country_count.get)
    return max_country
  Therefore, the final answer with python code is {print(solution())}
...More Lines...
Read the table blow regarding " + <title> + " to answer the following one question:
<Full Table>
<Question>
```

Figure 4: PoT implementation of applying Python Standard Library. Some lines are omitted due to the limited page.

```
Answer the questions based on the two given tables:

Rank | Cyclist              | Team            | Time     | UCI ProTour Points
1    | Alejandro Valverde (ESP) | Caisse d'Epargne  | 5h 29' 10 | 40
2    | Alexandr Kolobnev (RUS)  | Team CSC Saxo Bank | s.t.     | 30
3    | Davide Rebellin (ITA)    | Gerolsteiner      | s.t.     | 25
4    | Paolo Bettini (ITA)      | Quick Step        | s.t.     | 20
5    | Franco Pellizotti (ITA)  | Liquigas          | s.t.     | 15
6    | Denis Menchov (RUS)      | Rabobank          | s.t.     | 11
7    | Samuel Sanchez (ESP)     | Euskaltel-Euskadi | s.t.     | 7
8    | Stephane Goubert (FRA)   | Ag2r-La Mondiale  | + 2      | 5
9    | Haimar Zubeldia (ESP)    | Euskaltel-Euskadi | + 2      | 3
10   | David Moncoutie (FRA)    | Cofidis           | + 2      | 1

Question: who is the cyclist after Alexandr Kolobnev in the table?
Answer:
  def solution(table, header_dict):
    # Define the sub_table as a list of cyclist names
    sub_table = [row[header_dict["Cyclist"]] for row in table]
    # Find the cyclist after Alexandr Kolobnev
    kolobnev_index = sub_table.index("Alexandr Kolobnev (RUS)")
    if kolobnev_index != -1 and kolobnev_index < len(table) - 1:
      next_cyclist = table[kolobnev_index + 1]
      return next_cyclist
    else:
      return "Alexandr Kolobnev is either not in the table or is the last entry."
  Therefore, the final answer with python code is {print(solution(table, header_dict))}
Question: which country had the most cyclists finish within the top 10?
Answer:
  def solution(table, header_dict):
    # Define the sub_table as a list of cyclist names
    sub_table = [row[header_dict["Cyclist"]] for row in table]
    # Count the occurrences of each country in the top 10
    country_count = {}
    for cyclist_info in sub_table:
      country_start_index = cyclist_info.find("(") + 1
      country_end_index = cyclist_info.find(")")
      country = cyclist_info[country_start_index:country_end_index]
      if country in country_count:
        country_count[country] += 1
      else:
        country_count[country] = 1
    # Find the country with the most cyclists in the top 10
    max_country = max(country_count, key=country_count.get)
    return max_country
  Therefore, the final answer with python code is {print(solution(table, header_dict))}
...More Lines...
Read the table blow regarding " + <title> + "  to answer the following one question:
<Full Table>
<Question>
```

Figure 5: PoT implementation of applying Python Standard Library with parameters. Some lines are omitted due to the limited page.

```
Answer the questions based on the two given tables:

Rank | Cyclist              | Team            | Time     | UCI ProTour Points
1    | Alejandro Valverde (ESP) | Caisse d'Epargne   | 5h 29' 10 | 40
2    | Alexandr Kolobnev (RUS)  | Team CSC Saxo Bank | s.t.      | 30
3    | Davide Rebellin (ITA)    | Gerolsteiner       | s.t.      | 25
4    | Paolo Bettini (ITA)      | Quick Step         | s.t.      | 20
5    | Franco Pellizotti (ITA)  | Liquigas           | s.t.      | 15
6    | Denis Menchov (RUS)      | Rabobank           | s.t.      | 11
7    | Samuel Sanchez (ESP)     | Euskaltel-Euskadi  | s.t.      | 7
8    | Stephane Goubert (FRA)   | Ag2r-La Mondiale   | + 2       | 5
9    | Haimar Zubeldia (ESP)    | Euskaltel-Euskadi  | + 2       | 3
10   | David Moncoutie (FRA)    | Cofidis            | + 2       | 1

Question: who is the cyclist after Alexandr Kolobnev in the table?
Answer:
  #solution in Python
  import pandas as pd
  def solution(table_dict):
    df = pd.DataFrame(table_dict)
    # Find the index of Alexandr Kolobnev in the DataFrame
    kolobnev_index = df.index[df["Cyclist"].str.contains("Alexandr Kolobnev")][0]
    # Extract the cyclist after Alexandr Kolobnev
    next_cyclist = df.iloc[kolobnev_index + 1]["Cyclist"]
    return next_cyclist
  Therefore, the final answer with python code is {print(solution(table_dict))}
Question: which country had the most cyclists finish within the top 10?
Answer:
  #solution in Python:
  import pandas as pd
  def solution(table_dict):
    df = pd.DataFrame(table_dict)
    #since country in the column Cyclist, extract the country code from the column Cyclist
    df["Country"] = df["Cyclist"].apply(normalize_cyclist)
    # Count the number of cyclists from each country in the top 10
    country_counts = df["Country"].value_counts()
    most_cyclists_country = country_counts.idxmax()
    return most_cyclists_country
  #since country can not be obtained directly and is in "()" in column Cyclist, define the function which
  #can extract country from the Cyclist
  def normalize_cyclist(cyclist_info):
    import re
    # Regex pattern to extract content in ()
    pattern = r"\((.*?)\)"
    # Find the substring within parentheses
    match = re.search(pattern, cyclist_info)
    if match:
      return match.group(1)
    else:
      return ""
  Therefore, the final answer with python code is {print(solution(table_dict))}
...More Lines...
Read the table blow regarding ' + <title> + ' to answer the following one question:
<Full Table>
<Question>
```

Figure 6: PoT implementation of applying Python Pandas Library. Some lines are omitted due to the limited page.