WikiDT: Visual-based Table Recognition and Question Answering Dataset

Hui Shi[1], Yusheng Xie[2], Luis Goncalves[3], Sicun Gao[1], and Jishen Zhao[1]

[1] University of California San Diego
[2] Amazon AGI
[3] AWS AI Labs
{hshi, sicung, jzhao}@ucsd.edu, {yushx, luisgonc}@amazon.com

Abstract. Companies and organizations grapple with the daily burden of document processing. As manual handling is tedious and error-prune, automating this process is a significant goal. In response to this demand, research on table extraction and information extraction from scanned documents in gaining increasing traction. These extractions are fulfilled by machine learning models that require large-scale and realistic datasets for development. However, despite the clear need, acquiring high-quality and comprehensive dataset can be costly. In this work, we introduce the WikiDT, a TableVQA dataset with hierarchical labels for model diagnosis and potentially benefit the research on sub-tasks, *e.g.* table recognition. This dataset boasts a massive collection of 70,919 images paired with a diverse set of 159,905 tables, providing an extensive corpus for tacking question-answering tasks. The creation of WikiDT is by extending the existing non-synthetic QA datasets, with a fully automated process with verified heuristics and manual quality inspections, and therefore minimizes labeling effort and human errors. A novel focus of WikiDT and its design goal is to answer questions that require locating the target information fragment and in-depth reasoning, given web-style document images. We established the baseline performance on the TableVQA, table extraction, and table retrieval task with recent state-of-the-art models. The results illustrate that WikiDT is yet solved by the existing models that work moderately well on other VQA tasks, and also introduce advanced challenges on table extraction.

1 Introduction

Question answering is widely accepted as an AI-completeness task, while visual question answering (VQA) is an alternative to the visual Turing test [23]. VQA tasks, which require the integration of natural language and image understanding, attract tremendous interest from both computer vision and natural language processing communities. In general, VQA tasks can test a wide range of knowledge and inference skills, provided they can be related to information within an image. While knowledge in the wild world is extensive, VQA tasks typically bound the domain of the images and questions to make the task practical. General VQA tasks restrict the image domain to daily-life images with commonly seen objects, such as GQA[15] and VQA-v2[12]; Scene-Text VQAs confine their questions to text information on the images; and document VQAs ask questions on the image-formed documents, for example, OCR-VQA[26], DocVQA[25], and VisualMRC[41].

While most VQA tasks remain primarily in the research domain, document VQA demonstrates significant commercial value for machine-learning-as-a-service providers, generating tangible revenue. Its potential is reflected in the growing demand for intelligent document processing (IDP), with the global market size expected to surge 370% from 1.1 billion USD in 2022 to 5.2 billion in 2027 [1]. IDP solutions offer compelling benefits that reduces manual labor. For instance, insurance companies could use IDP to automatically extract the per-item cost of a claim from user-scanned receipts at a massive scale, significantly reducing the human labor needed. Moreover, tax preparation software could leverage IDP to handle diverse types and formats of income reports and generate tax return documents for a vast customer base.

The research on Document VQA, or IDP, faces several challenges despite its growing demand. Firstly, the differences in the image and questions types between existing document VQA datasets and the real-world applications poses an obstacle. Current dataset often focus on extractive questions with limited candidate answers, while real-world applications usually either have long context or require an answer that can not directly extract from the text. For example, only about 500 out of 30,000 samples in InfographicVQA[24] are non-extractive, which require reasoning or synthesis beyond simply locating information within the image. Even the extractive samples rely on short OCR context compared to the length of actual documents. Secondly, while the diversity of document and element structure is crucial for real-world applications, achieving it in datasets is significantly challenging. Collecting data from a single source could results in highly similar samples, while combining data from difference sources involves additional data cleaning, verifying and unifying procedures. Overcame the challenges in document collection, obtaining annotation from human is costly and can introduce inconsistencies and inaccuracies. Finally, the absence of intermediate labels in the datasets hinders developing models that can generalize to unseen tasks effectively.

Development of well-annotated and diverse dataset is essential for advancing the document VQA, observing that existing dataset's limitations are hard to overcome by current models alone. Recent research in model design highlights the urgent need for fully annotated datasets that explicitly incorporate chains of reasoning from context to answer. The specialization benefit is obvious, despite the prosperity of recent unified models like T5[33] and GPT-3 [5]. For instance, a unified spatial aware text-to-text model for general document VQA can achieve 39.32 % accuracy [4], while a table-specialist model can achieve 50.97 % accuracy on the same dataset [14]. On the other hand, though end-to-end labeling is cheap to obtain, recent studies show that the intermediate labels or chains of reasoning steps can critically improve the performance and the robustness of the models [35,44,45,37].

To this end, we create a document VQA dataset that specialized in Table QA with abundant intermediate labels, named WikiDT (Wikipedia Document Table). WikiDT is closer to real-world applications that processing table-present documents. The documents in the dataset contain tabular information and natural language questions. The task requires the model or the system to comprehend the tabular structure, retrieve pertaining information, and perform SQL-like operations, *e.g.* filtering, count, and summation. Aware of the challenges introduced in our tabular document VQA task, we provide intermediate labels to each of the sub-tasks: table extraction, table retrieval, and question

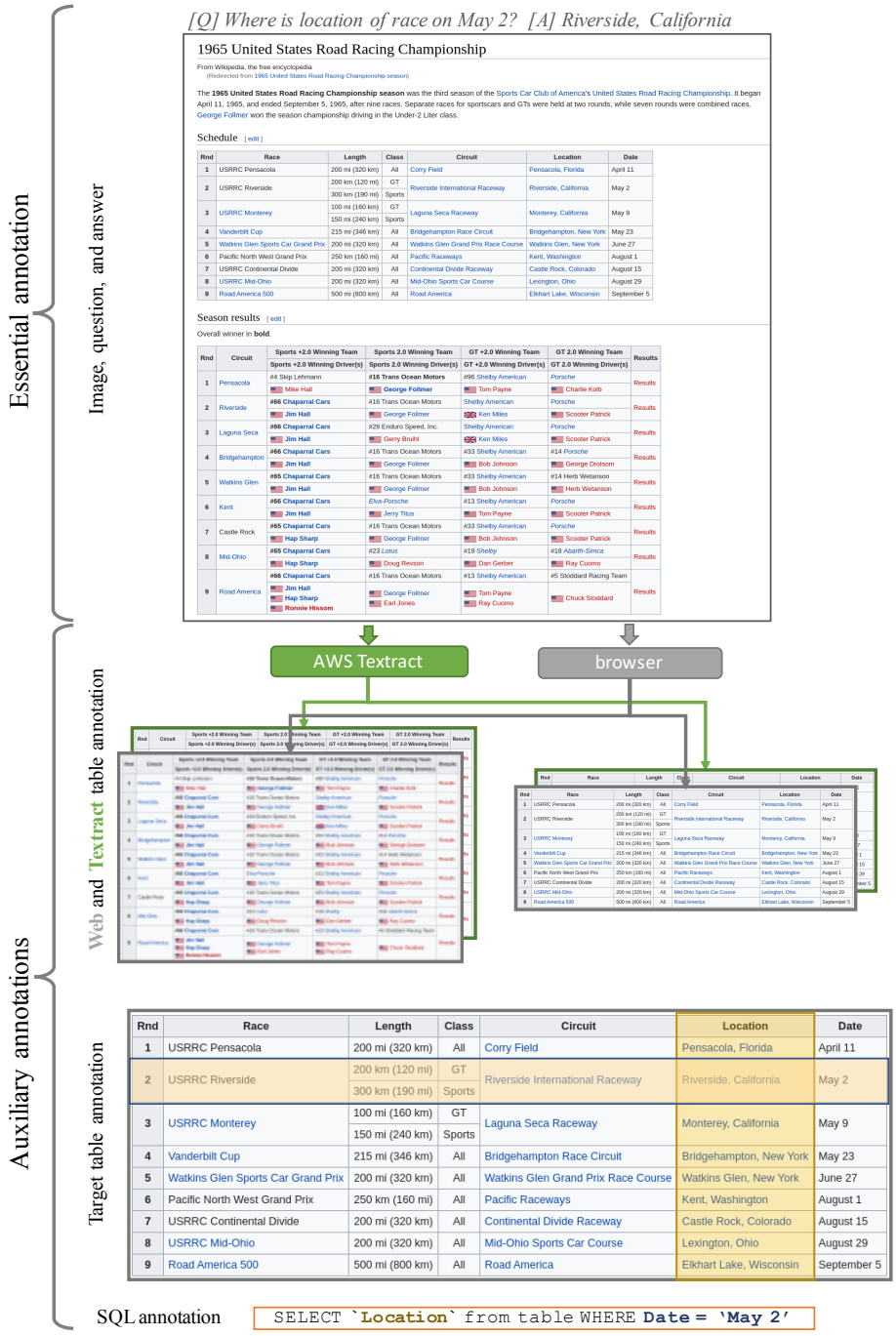


Fig. 1: Example of labels in WikiDT dataset.

answering. Overall, WikiDT dataset contains 16,887 images, 159,905 tables and their annotations, and 70,652 question-answer pairs with table retrieval labels. Lastly, the experimental results demonstrate that all the sub-tasks in the WikiDT dataset introduce uncovered challenges to the existing models, and the challenges reveal directions for improvement of model designs in the future.

The WikiDT dataset offers the following unique features:

- A specialized table VQA dataset with adequate samples and accurate intermediate labels.
- Unique VQA reasoning challenges, including multi-level span prediction, multi-step reasoning, and a diverse and exponential answering space.
- Multi-level ground-truth labels that facilitate diagnosis of the end-to-end model and enable training for sub-tasks like table recognition and retrieval.
- Challenges in table recognition, such as multiple header rows/columns, and diverse table size and position.

2 Related Work

In this section, we relate WikiDT to other visual question-answering categories and classify WikiDT as a specialized task under the document VQA. Furthermore, WikiDT is compared to the TableQA and table recognition tasks, hightlighting its unique diversity and challenges.

VQA task categories. VQA task is generally defined as question answering based on any image-type input. Typically the questions are about objects and their attributes, *e.g.*, object type, shape, color, and texture, as well as their spatial relationships. CLEVR[17], GQA[15], VQA-v2[12] are such general VQA datasets. A new research direction has emerged, focusing on answering questions about textual information within images. For example, given a picture of a grocery store interior, this research aims to answer questions such as "what's the price of the product in the center of the picture?" As images in these datasets are usually natural scenes with text, this type of the VQA task is named Scene-Text VQA. The datasets including *e.g.*, TextVQA[39], ST-VQA[3] and EST-VQA[44]. In Scene-Text VQA, the textual information is unstructured, and one of the key capability requirements is to pair the textual information to the spatial information of objects, *e.g.*, price of good, texts shown in the center. On the contrary, when the texts on the images are structured, *e.g.*, images that have tables and charts, which are commonly seen in documents, books, and websites, the task is usually referred to as Document VQA and emphasizes on understanding the structured information. An example question could be "what is the largest value in a list?" With the diversity of chart types, it's less practical to build a monolithic model that comprehends all kinds of infographics and performs different types of reasoning. Among the charts, tables stand out as a particularly rich source of information. Therefore, we proposed WikiDT, a large-scale document VQA dataset that specialized in TableVQA.

Figure 2 shows the categorization of the VQA tasks with typical datasets for each category, and Table 2 compared the datasets in detail.

The table VQA task in DUE[4] is the most similar existing task to WikiDT. However, DUE presents a reduced challenge as its QA context is an image that contains only the

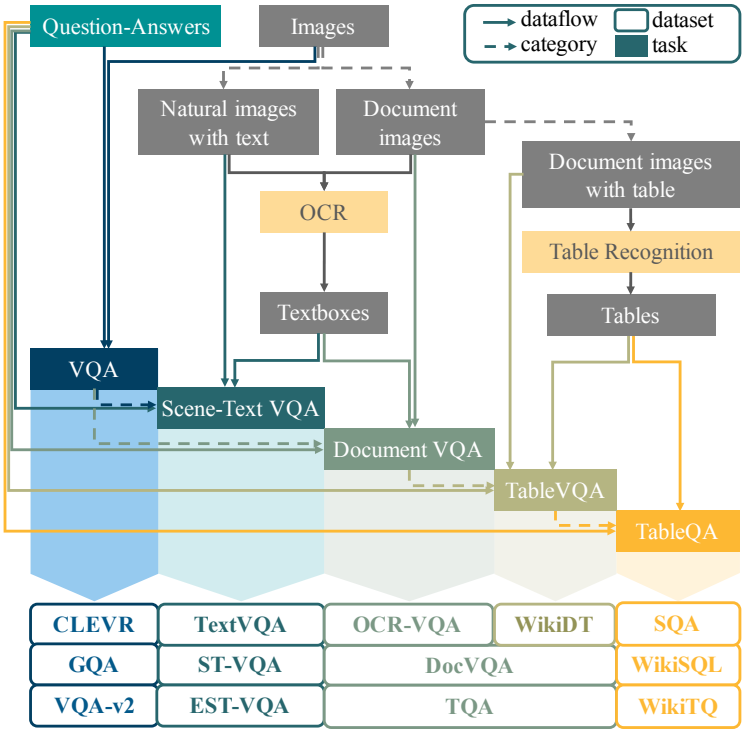


Fig. 2: Task categorizations and relationships.

target table. In contrast, WikiDT, while similar in question and answer length to DUE and other scene-text VQA and DocVQA datasets, contains significantly more textual information, as shown in Table 6. This extended textual context makes WikiDT more realistic and challenging.

TableQA and TableVQA. Though WikiDT is created from the TableQA dataset, it's not simply a TableQA task that replaces the table with an image. WikiDT is different from its ancestors (*e.g.*, WikiSQL[46] and WikiTQ[31]) in two ways: 1) mainstream of the TableQA tasks focus on semantic parsing, a task translating questions in the natural language into the executable SQL query, where the table schema is definite and strict. TableVQA does not assume the availability of table schema and the rigorousness of the language; 2) WikiDT also requires the model to recognize and identify the table from a noisy document with all other related information, which is more similar to the application of automated document processing.

A similar attempt to WikiDT is DUE[4], which formulates its TableVQA by using the same WikiSQL dataset and replacing the table with an image with only the table (exampled in Figure 13). In Section 4, the results show that giving a noiseless image that contains only the target table makes the TableVQA trivial since the table structure extraction from that images are well-performed. Moreover, contrary to DUE that each image is a single table, images in WikiDT may contain multiple tables.

Dataset	Annotations	Size
ICDAR-13[11]	TD/TSR	0.45k
ICDAR-19[10]	TD/TSR	3.2k
Marmot[4]	TD	2k
SciTSR [8]	TD/TSR	15k
PubTabNet[47]	TSR	568k
TableBank[20]	TD/TSR	417k
PubTables-1M [40]	TD/TSR	948k
WikiDT-detection	TD	54,032
WikiDT-structure	TSR	159,905

Table 1: Comparison of table extraction dataset.

Comparing to existing table recognition dataset. Labeling for the table structure recognition dataset is labor intensive and imprecise. Even a 5-row by 5-column small table requires drawing 25 bounding boxes for each of the cells, and the cell boundary is usually just a few pixels from its neighbors. Therefore, recent large-scale table recognition datasets, and also WikiDT, are created from an automated process: given the source of Latex/Word/HTML, we render the source to images meanwhile extract the bounding boxes and texts from the rendering software. When using the dataset for training, the models are only provided with images and bounding box labels, so they could not hack the problem from the source. In addition to diagnostic labels to our TableVQA task, the table recognition labels also would benefit the table recognition research with versatile table shapes and layouts. With the continuous scroll feature of the web pages, compared to PDFs and Words, WikiDT contains tables that do not fit into a single print paper and can be leveraged as a benchmark for table recognition when the table spans multiple pages. An overall comparison to the table recognition dataset is shown in Table 1, and a detailed comparison to the PubTables-1M can be found in Figure 16 and 17.

3 Dataset Description

The WikiDT dataset contains 16,887 documents with 70,652 QA samples. Each question can be answered from a single document, and the answer should be inferred from one of the tables on the document. As a common practice in Scene-Text and Document VQA, the OCR result on the page is made available to the user. For table VQA model, the input is the document as an image, texts, and bounding boxes of all textual information on the image, and the output should be the answer to the question.

Besides essential input (image and question) and output (answer) for the TableVQA task, WikiDT also provides auxiliary annotation that helps solve the TableVQA task step-by-step. Illustrated in Figure 1, intuitively, human fulfills the table VQA task by: recognizing the tables, identifying the table that relates to the question, then reasoning on the table to find the answer. To facilitate the diagnosis of end-to-end model and the training of modules in ensemble systems, WikiDT dataset also provides the following intermediate labels as auxiliary annotations:

[4] https://www.icst.pku.edu.cn/cpdp/sjzy/

Dataset	Image	Auxiliary input	Metric	Count (QA/Document/Tables)
CLEVR*†[17]	synthetic	logic structure	accuracy	850k/85k/-
GQA†[15]	natural	scene graph	accuracy	22M/110k/-
ST-VQA[3]	natural	OCR	ANLS	32k/23k/-
TQA[19]	textbook	OCR, topic graph	accuracy	26k/1k/-
VisualMRC[41]	webpage	ROI and relevance label	BLEU, etc.	30k/10k/< 1k
DocVQA[25]	documents	OCR	ANLS	50k/13k/11.8k
InforgraphicQA[24]	documents	OCR	ANLS	30k/5.4k/9.5k
DVQA*[18]	synthetic bar charts	chart metadata	accuracy	3.4M/300k/-
DUE-TableVQA[4]	cropped tables	-	accuracy	120k/16k/16k
WikiDT	webpage	various	accuracy	70k/17k/160k

Table 2: VQA dataset comparisons. † denotes classification tasks. * marks the synthetic datasets that generate the QA and images. ANLS represents Average Normalized Levenshtein Similarity; VisualMRC uses image captioning task metrics including BLEU[30], METEOR[9], ROUGE-L[21], CIDEr[42]; DUE-TableQA and WikiDT use denotation accuracy.

- Table annotations on both full-length and paged images.
- OCR and AWS Textract table recognition results on paged images to simulate the real-world scenarios.
- Table retrieval labels indicating which table corresponds to each question.
- SQL queries that generate answers from tables.

WikiDT is available at https://huggingface.co/datasets/AmazonScience/WikiDT. Data acquisition and processing are detailed in the supplementary materials.

3.1 Data Acquisition

WikiDT is extended from the WikiSQL[46] and WikiTableQustions(WikiTQ)[31]. The images are rendered from the URL in WikiSQL and WikiTQ's metadata by a browser that also generates the raw table annotations at the same time. The questions in WikiDT are combined from WikiSQL and WikiTQ, in which the questions are all human-annotated. For the questions from the WikiTQ dataset, we directly use the original human-labeled answers, and for the questions from the WikiSQL dataset, we generate answers by executing their human-labeled SQL query on the aforementioned table annotations. Since multiple tables may appear on the same image, we also semi-automatically labeled the target table to the question. While the browser generated *oracle* table annotation is great for decoupling the tasks and diagnosing the end-to-end model, they are assumed unavailable in real-world scenarios. Therefore, following the existing practice, we also provide the OCR and table recognition results on the images from the publicly available service, AWS Textract.

Task	Input	Output
Table Detection	image	table bbox(es)
Table Structure Recognition	image	row, column, cell bboxes
Table Retrieval	tables, question	target table
TableVQA	image, question	answers

Table 3: Sub-task formulations.

Page	Min	Mean	Max	Variance
Full page	1200	6592	78423	6061
Subpage	15	1761	57622	1028

Table 4: Image size (in pixel) statistics.

3.2 TableVQA and sub-tasks

The multi-level intermediate labels allow many possibilities for diagnosing the TableVQA models and training modules for sub-tasks. In this paper, we evaluate TableVQA and two sub-tasks, table recognition and retrieval. Table 3 summarizes the input and output for each (sub-)task.

Table recognition task encompasses two sub-tasks: **table detection(TD)** and **table structure recognition(TSR)**. Table detection involves predicting the table regions within images, typically annotated as rectangular bounding boxes. Following the practice established by PubTables-1M, table structure recognition data is provided as pairs of cropped images and structure annotation. Each image crop contains only a single table with consistent padding, while the structure annotation details the bounding boxes of rows, columns, cells (including multi-row and multi-columns cells, hereinafter, merged cells), and table headers.

Table retrieval aims to identify the specific table within an image that can answer a given question. The retrieval model should not peek at the answer.

TableVQA should answer the question given the full-page image. This end-to-end solution remains a significant challenge for existing models. To simplify this task, we explore leveraging the auxiliary annotation, such as restricting the context OCR to only the table regions.

3.3 Data Analysis

The statistics of the WikiDT are described as follows.

Images and tables. The *full page* images refer to the original auto-scrolled web screenshots, which usually are long and hard for existing visual models to take as input. A pagination heuristic is applied to segment the full-page screenshot into small segments without cutting through tables and texts. The segmented images are referred to as *subpages*. Table 4 lists the heights of the full pages and subpages images, which re-emphasizes the motivation of the pagination process.

Questions and answers. Figure 3 shows the type and number of items in the answer, as well as the cumulative probability of answer strings. Most questions in WikiDT expect

Data	Count
Images (full page)	16,887
Sub-page images	54,032
Table annotations	159,905
Question-answer pairs, retrieval labels	70,652
SQL annotations	49k
Single answer questions	68,573
Multiple answer questions	3,524
Mean number of tables per full page	14.7 (21.9)
Mean number of tables per sub-page	3.4 (16.0)
Mean number of words per sub-page	708.9 (517.1)
Mean number of words per table	170.84 (458.7)
Mean number of columns per table	9.08 (48.3)
Mean number of rows per table	12.15 (18.1)

Table 5: Basic statistics on WikiDT. Standard deviations are shown in the brackets.

Dataset	Question	Answer	Image
ST-VQA	8.8	1.6	7.5
TextVQA	8.1	1.5	12.2
DocVQA	9.5	2.4	182.8
InfographicVQA	11.5	1.6	217.9
WikiDT	11.0	2.0	708.9 (subpage)

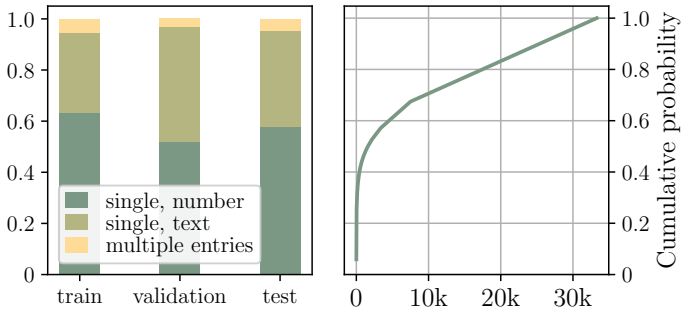
Table 6: Average lengths in number of words.

a single answer, and roughly only 5% answers have more than one entry. A single answer can be a phrase, a word, or a number, while a multi-entry answer is like {*United States*, *Canada* }. For multi-entry answer, denotation accuracy requires the model to answer correctly all the entries, regardless of the order. Despite most samples only expect a single entry answer, the diversity of answer suggests the WikiDT TableVQA task should be solved in a generative way. In datasets that can be formulated as classification tasks, such as CLEVR and GQA, the top 1000 answers could cover more than 90% of questions. In the contrary, 1000 most frequent answers could only cover 70% samples.

Social impact. WikiDT provides diagnostic data that potentially advances multiple machine intelligence areas. The curation of the dataset involves trivial privacy concerns because the data source of WikiDT is Wikipedia pages, where the user, the editor, information is not revealed, and the contents of the pages are publicly available. The potential negative impact of the WikiDT is two-fold: misinformation and outdated information intrinsically from Wikipedia, and the bias towards certain countries (*e.g.*, the *United States* and *Canada* have a much higher frequency than other country names).

4 Experiments

In this section, we evaluate existing neural networks on TableVQA, table recognition, and table retrieval tasks.



(a) Answer type and items (b) Top-n answer strings

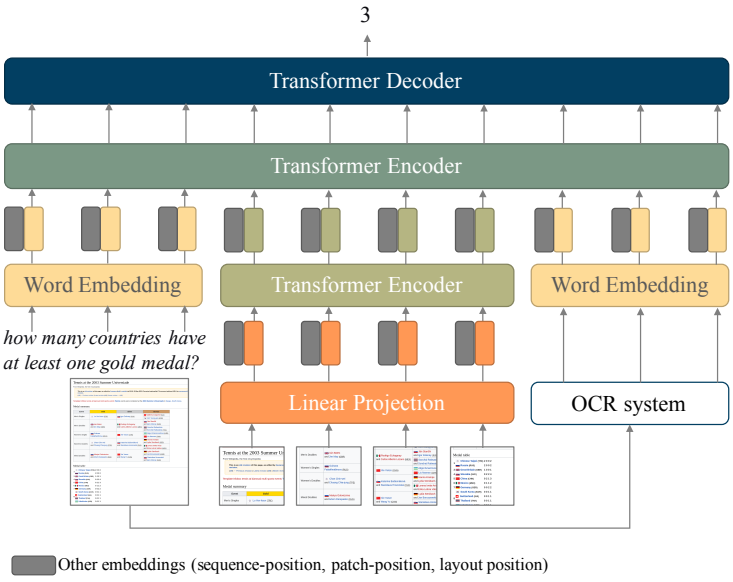
Fig. 3: Analysis of answer items/types and cumulative word probability.

4.1 TableVQA.

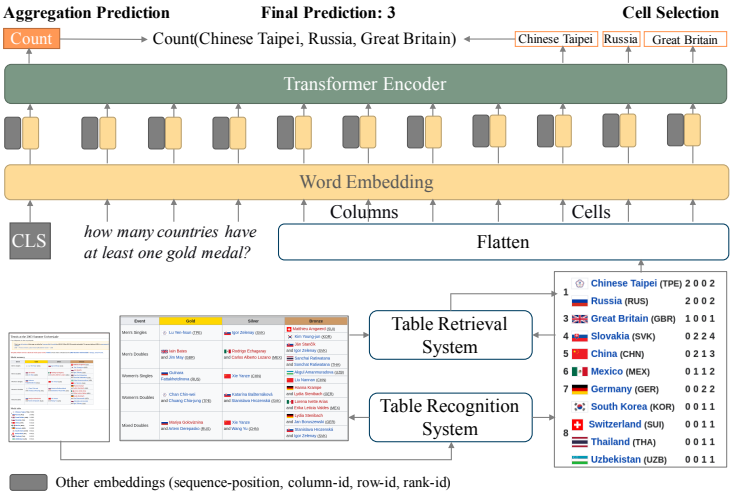
Task setup. In typical model setup, the entire image and all OCR results are used as input. However, in this work, we restrict the input context by leveraging auxiliary labels. As shown in Table 5, the number of words per subpage image can easily exceed a thousand, and this number further increases after tokenization into subwords. Additionally, a question might associate with more than one subpage, potentially resulting in thousands of tokens, which overload the computation resources. To make the task attackable, we utilize the table detection results and table retrieval labels, such that we can prune irrelevant content from the input context.

Models. The competition between end-to-end and modularized models is yet concluded, so as in the VQA domain. Thus we explore the state-of-the-art performance under the two approaches: LaTr[2] and T5 [33] model as end-to-end representatives, and TAPAS[14] with table recognition module and ground table retrieval labels as the modularized approach. The high-level architectures of the two models are shown in Figure 4.

– **Modularized model** generally should consist of three components of table recognition, table retrieval, and table question answering. In this experiment, we utilized the auxiliary annotation in WikiDT to substitute the first two modules. The table recognition outcome is from Textract. To the authors' knowledge, there are no table retrieval models for the table question answering is open-sourced, thus we skip this step and use the ground-truth retrieval labels given the Textract tables. The table question answering module is the state-of-the-art TAPAS [14] model. As shown in Figure 4, the TAPAS model takes the tokenized question, a special token *CLS*, and the flattened table as the input. A flattened table includes the names of columns and the content in each table cell. Each token content is embedded by an embedding layer, and the embeddings are then concatenated with extra learnable embeddings for the position in the sequence, column and row ID, rank, and the token position in the table cell. After the transformer encoder, each token has an output vector. The output vector of *CLS* is fed into a classifier to predict the aggregation operation (*e.g.*, None, Count, Sum, Average), the column name outputs will decide the column to be selected, and



(a) LaTr architecture



(b) TAPAS architecture

Fig. 4: TableVQA model architectures.

the table cell output in the selected column will independently predict if the cell is selected. Lastly, the aggregation is applied to the selected cells and produces the final prediction.

– **Monolithic model:** We compare two similar neural architecture: T5[33] and LaTr[2]. Both of them receive the question on a subpage that contains the target table, and the OCR results that locate within the region of the target table. The monolithic models are not informed by the table recognition results. LaTr model architecture is shown in Figure 4 on the top. The input sequence contains the tokenized question, patched and flattened image of the content, and the OCR results on the image. Before feeding into the transformer encoder, the extra encodings of the layout position, token position, and patch position are concatenated to the corresponding embeddings. Unlike the TAPAS model, the LaTr directly generates the answer with a transformer decoder. While the LaTr knows the location of each OCR token, the T5 model is inherently spatially-blind. We use the vanilla T5 model in which the input to the encoders contains only the textual (word embedding) and the sequence order information (positional embedding). Despite T5 knows no spatial information, as table tokens are fed into model by row and column order, T5 can potentially understand the table in this linearized form.

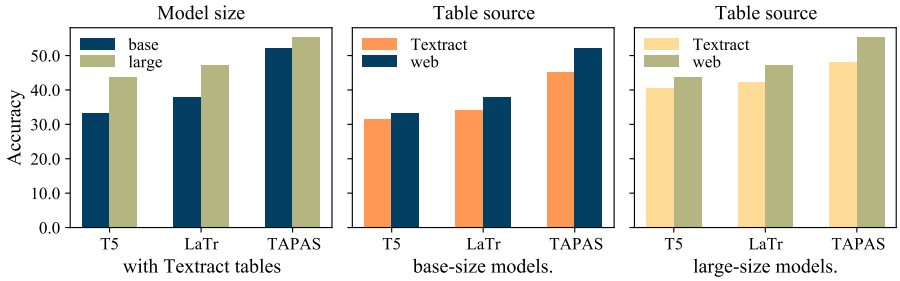


Fig. 5: TableVQA model diagnosis. TAPAS model is more sensitive to table recognition error, and T5/LaTr models are more sensitive to model size. Also, the performance of TAPAS with web tables set the upper bound accuracy of improving table recognition.

Implementation. We finetuned the TAPAS and T5 from their pre-trained models released on Huggingface. For LaTr, we shared the WikiDT with its authors and received the results after fine-tuning the pre-trained LaTr.

Metrics. The performance on Document TableQA is evaluated by denotation accuracy, detailed in [27]. The prediction is correct if 1) it has the same number of entries as in the ground truth answers; 2) every entry in the prediction can non-repeatedly match to an entry in the ground truth answers (ignoring the format variances, *e.g.* 1000 *v.s.* 1,000).

Results. Table 7 summarizes the model performances. TAPAS model achieves the best performance unsurprisingly as it receives the table structure information and is designed to tackle such table-operation tasks. LaTr outperformed the T5 model with only marginal advantage, even the LaTr is visual- and spatial-aware and is pre-trained heavily on the scene-text VQA and document VQA tasks but T5 is only pre-trained on the text-to-text tasks. The results indicate that enhancing document VQA models with table-specific designs is the way to improve their performance in table-based reasoning.

Model	Denotation Accuracy (%)		
	Single Answer	Multi-answer	Overall
T5	32.74	1.30	31.67
LaTr	35.29	0.0	34.08
TAPAS	46.24	6.86	45.23

Table 7: Model performance on WikiDT TableVQA task.

Diagnostic and ablation experiment. Owing to the auxiliary annotations, we can analyze the error origins. Figure 5 compares the performance when using the web (ground truth) table annotation against using the Textract annotation, and the impact of model sizes. For monolithic models, the performance improvement with more accurate table annotation is trivial compared with the benefit of increasing the model size, which is contraray to the conclusion on the modularized model of TAPAS.

4.2 Table Extraction.

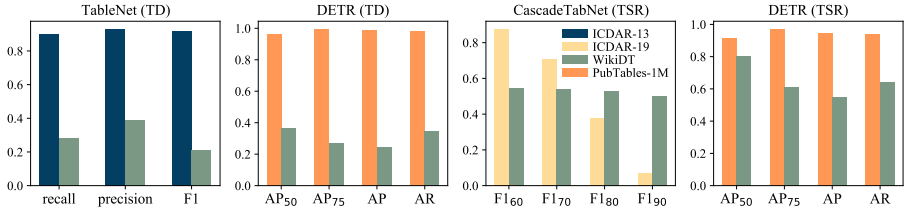


Fig. 6: Performance of pretrained model on WikiDT table recognition. Each pair of bar shows the model's performance difference between its pre-training dataset (ICDAR-13, ICDAR-19, or PubTables-1M) and inference dataset (WikiDT).

The table extraction experiment aims to show that the documents in WikiDT offer different challenges and are complementary to the existing datasets. We demonstrate by 1) evaluating the table extraction performance on models that are trained on other datasets, and showing the model performance is compromised owing to significant domain difference; 2) comparing the performance of the model trained on WikiDT and PubTables-1M dataset and showing that WikiDT contains more challenging table layouts.

Models and configurations.

- TableNet[28] is a CNN using pretrained VGG[38] or DenseNet[16] as encoder and has two up-sampling CNN decoders to predict the table mask and column mask. We use TableNet pretrained on the Marmot dataset with DenseNet-121 as the encoder.
- CascadeTabNet[32] also adopts CNN. Its encoder is a pretrained HRNetV2p-W32[43], and the decoder uses cascadeCNN-like architecture to predict borderless table mask, cell masks, and bordered table mask. Post-processing uses line detection to translate

pixel-level masks to bounding boxes. We evaluate the CascadeTabNet pretrained on ICDAR 19.

– DETR[6] is a transformer neural network for object detection. In table detection and structure analysis, tables, rows, columns, and table headers are considered as distinct types of objects. We evaluate DETR pretrained on PubTables-1M, and with the same model configuration, we train another DETR from random initialization on WikiDT.

Metrics. Generally, the evaluation metric for TD and TSR is IoU-based precision and recall[22], *e.g.*, Average Precision(AP_{IoU}), Average Recall(AR_{IoU}) [5] and F1 scores.We report the metrics used in evaluating the dataset that each model is trained on. The TableNet predicts binary masks instead of bounding boxes, hence the precision, recall, and F1 are reported on the pixel level.

Task	Table detection				Table structure recognition			
Dataset	AP_{50}	AP_{75}	AP	AR	AP_{50}	AP_{75}	AP	AR
PubTables-1M	96.6	99.5	98.8	98.1	91.2	97.1	94.8	94.2
WikiDT	84.7	72.6	67.5	72.7	88.9	79.6	74.7	82.6

Table 8: DETR performance on PubTables-1M and WikiDT

Dataset difficulty. Table 8 shows the performance of DETR, the current SOTA model for table recognition, on the PubTables-1M and WikiDT datasets. DETR achieved significantly lower performance on the WikiDT dataset, indicating that WikiDT is challenging compared to the existing largest table recognition dataset, in both table detection and table structure recognition. The complexity comes from both the flexibility of table location in the page, table shape, and inner structures.

Domain transfer. Figure 6 presents the performance gap between models on their pre-trained dataset and WikiDT. The results show that WikiDT exhibits substantially different layouts and table styles that models trained on the existing dataset could not generalize to. Especially, the performance decrease more severely in the table detection (TD) than in the table structure recognition, which indicates larger difference in document layout and table size than in table style.

Conclusions. The experiments illustrate the WikiDT provides a challenging table recognition dataset that is substantially different from the existing datasets in page layout and table styles. We refer the readers to the supplementary material for a detailed comparison and qualitative examples.

4.3 Table Retrieval

The WikiDT-retrieval task requires retrieving the table relevant to the question from the collection of tables on the same webpage. Viewing from Table 5, in most cases, for each query (question) the candidate tables number is small, especially compared to retrieval

[5] the IoU in the subscript is in 10^{-2}, e.g. AP_{75} is the average precision under IoU threshold of 0.75.

Method	Web Tables	Textract Tables	Diff.
BM25	0.382	0.389	-0.007
Dense Retrieval	0.587	0.524	0.063

Table 9: Table retrieval performance in Mean Reciprocal Rank (MRR), which is computed on the subset of samples with multiple table candidates.

in a recommendation system or search engine[29,36]. However, the challenge is that the tables from the same webpage might have closely related contents that are hard to distinguish.

Dataset split. The dataset split is the same as other tasks. In the development and test set, the samples having only one table are removed. In the training set, if a training sample contains only a single table, we additionally sample two background tables from all the tables in the dataset. Otherwise, the background tables to a question are other tables on the same page.

Models. We evaluate BM25[34] and a dense retrieval method on table retrieval task.

- BM25: questions are queries and flattened tables on the entire page are a corpus.
- Dense retrieval: we use the pre-trained TAPAS model with an additional special token R_CLS to predict the likelihood of if the given table is the target table. The dense retrieval method is trained with weighted binary cross-entropy loss. Due to the intrinsic label imbalance, a sample weight of $|target\_sample| \,/\, |background\_sample|$ is applied to all the background samples.

The results are summarized in Table 9. The dense retrieval model significantly outperformed the non-learning method, BM25. Similar to the QA task, the inaccuracy in the table recognition (Textract tables) has a marginal impact on the retrieval results.

5 Conclusions

This paper introduces WikiDT dataset, whose primary goal is to facilitate visual-based table question answering. Featuring multi-modality data and rich intermediate labels in WikiDT, the dataset serves a variety of tasks, encompassing table extraction, table retrieval, and TableVQA. The evaluation of the recent models on each task demonstrates the WikiDT imposes novel challenges that are yet solved. Especially, the current document VQA or scene-text VQA models have great difficulty to solve TableVQA questions that require multi-step reasoning. Proposals for advanced models and approaches to address these challenges are deferred to future work.

6 Acknowledgement

Research reported in this publication was supported by an Amazon Research Award, Fall 2022 CFP, and Amazon Post-Internship Fellowship. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of Amazon.

References

1. Intelligent document processing market size and forecast. https://www.verifiedmarketresearch.com/product/intelligent-document-processing-market/. Accessed: 2022-10-30. 2
2. Ali Furkan Biten, Ron Litman, Yusheng Xie, Srikar Appalaraju, and R. Manmatha. Latr: Layout-aware transformer for scene-text vqa. In *CVPR 2022*, 2022. 10, 12
3. Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusinol, Ernest Valveny, CV Jawahar, and Dimosthenis Karatzas. Scene text visual question answering. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4291–4301, 2019. 4, 7
4. Łukasz Borchmann, Michał Pietruszka, Tomasz Stanislawek, Dawid Jurkiewicz, Michał Turski, Karolina Szyndler, and Filip Graliński. Due: End-to-end document understanding benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. 2, 4, 5, 7
5. 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. 2
6. 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. 14
7. 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. 19
8. 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. 6
9. Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, pages 376–380, 2014. 7
10. 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. 6
11. 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. 6
12. Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017. 1, 4
13. 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. 19
14. 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. 2, 10
15. Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 4, 7

16. Forrest Iandola, Matt Moskewicz, Sergey Karayev, Ross Girshick, Trevor Darrell, and Kurt Keutzer. Densenet: Implementing efficient convnet descriptor pyramids. *arXiv preprint arXiv:1404.1869*, 2014. 13
17. Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017. 4, 7
18. Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. Dvqa: Understanding data visualizations via question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656, 2018. 7
19. Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In *Proceedings of the IEEE Conference on Computer Vision and Pattern recognition*, pages 4999–5007, 2017. 7
20. Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou, and Zhoujun Li. Tablebank: A benchmark dataset for table detection and recognition. *arXiv preprint arXiv:1903.01949*, 2019. 6
21. Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81, 2004. 7
22. 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. 14
23. Sruthy Manmadhan and Binsu C Kovoor. Visual question answering: a state-of-the-art review. *Artificial Intelligence Review*, 53(8):5705–5745, 2020. 1
24. Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706, 2022. 2, 7
25. Minesh Mathew, Ruben Tito, Dimosthenis Karatzas, R. Manmatha, and C. V. Jawahar. Document visual question answering challenge 2020, 2020. 1, 7
26. Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 947–952. IEEE, 2019. 1
27. Will Monroe and Yushi Wang. Dependency parsing features for semantic parsing, 2014. 12
28. 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. 13, 26
29. Feifei Pan, Mustafa Canim, Michael Glass, Alfio Gliozzo, and Peter Fox. Cltr: An end-to-end, transformer-based system for cell level table retrieval and table question answering. *arXiv preprint arXiv:2106.04441*, 2021. 15
30. Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002. 7
31. Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. *arXiv preprint arXiv:1508.00305*, 2015. 5, 7, 19
32. 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. 13
33. Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. 2, 10, 12

34. Stephen Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. In *Overview of the Third Text REtrieval Conference (TREC-3)*, pages 109–126. Gaithersburg, MD: NIST, January 1995. 15
35. Hui Shi, Sicun Gao, Yuandong Tian, Xinyun Chen, and Jishen Zhao. Learning bounded context-free-grammar via lstm and the transformer: Difference and the explanations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8267–8276, 2022. 2
36. Hui Shi, Yupeng Gu, Yitong Zhou, Bo Zhao, Sicun Gao, and Jishen Zhao. Every preference changes differently: Neural multi-interest preference model with temporal dynamics for recommendation. *arXiv preprint arXiv:2207.06652*, 2022. 15
37. Hui Shi and Yang Zhang. Deep symbolic superoptimization without human knowledge. *ICLR 2020*, 2020. 2
38. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 13, 19
39. Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 4
40. Brandon Smock, Rohith Pesala, Robin Abraham, and WA Redmond. Pubtables-1m: Towards comprehensive table extraction from unstructured documents. *arXiv preprint arXiv:2110.00061*, 2021. 6
41. Ryota Tanaka, Kyosuke Nishida, and Sen Yoshida. Visualmrc: Machine reading comprehension on document images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13878–13888, 2021. 1, 7
42. Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575, 2015. 7
43. 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. 13
44. Xinyu Wang, Yuliang Liu, Chunhua Shen, Chun Chet Ng, Canjie Luo, Lianwen Jin, Chee Seng Chan, Anton van den Hengel, and Liangwei Wang. On the general value of evidence, and bilingual scene-text visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 2, 4
45. Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022. 2
46. Victor Zhong, Caiming Xiong, and Richard Socher. Seq2sql: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103*, 2017. 5, 7, 19
47. Xu Zhong, Elaheh ShafieiBavani, and Antonio Jimeno Yepes. Image-based table recognition: data, model, and evaluation. *arXiv preprint arXiv:1911.10683*, 2019. 6

A Data Acquisition and Processing

A.1 Overview

WikiDT is created from Wikipedia, a public online encyclopedia created and updated by its users. We are only interested in those Wikipedia web pages with tables, for purpose of table extraction and table-based question answering. Overall, the data acquisition process is taking the Wikipedia URLs from existing datasets, namely WikiTableQuestions[31], TabFact[7], and WikiSQL[46], rendering the web pages to images while annotating the tables, then re-connecting the rendered images and table annotations to the existing question-answer pairs.

The basic annotations in the WikiDT are {image, question, answer} triplets, which are basic input and output to the end-to-end visual-based table question answering models. Aware of the challenges, WikiDT also provides auxiliary annotations include: ground truth table annotations (web tables); Textract table recognition and OCR annotations; QA target table labels based both on the Textract table and web tables; and executable SQL queries on majority of QA samples.

Although the task may seem straightforward, the creation of the dataset is fraught with numerous challenges. Especially, the web contents has changed dramatically since the original WikiSQL dataset was created. Additionally, the data processing in WikiSQL dataset is unknown and irreversible, which poses great hardship in re-connecting the tables to the question-answer pairs. We detail the approaches to overcome those as below.

A.2 Image rendering and pagination

We leverage the Puppeteer[6] to render the screenshots mainly from the URLs. Generally, the rendered images in the continuous scroll mode have large heights that are extremely hard for the popular visual networks to handle. Thus, the original rendered pages (*full-page*) are paginated to several *sub-pages*.

URLs. TabFact provides the URLs to the tables in itself and WikiSQL, and WikiTableQuesions also provide the URLs to its tables. Despite all the URLs are Wikipedia pages, the URLs in the TabFact is Wikipedia domain followed by article title, which changes drastically over the years, while the WikiTableQuestions provided URLs with archived version IDs, which have the exact content when they created the dataset. So for the first step, we search for the Wikipedia editing history for each TabFact pages, and find the closest version to then the WikiSQL dataset is created. Then both of the recovered TabFact pages and WikiTableQuestions pages are rendered.

Renderer setting. The width of the page is set to 1600 pixels for a better table layout (as compared in Figure 9). Less than 0.5% of pages have minimum page width requirements that are larger than 1600 pixels, mainly owing to extremely wide tables, and are rendered with their minimum page width.

Pagination. The extra-long full pages cause huge trouble for the down-stream tasks. For example, the typical width-height ratio of input layer to common visual models is 1:1 for VGG[38] and ResNet[13], and other backbone models trained on ImageNet. Therefore the pagination is critical for usage of the dataset. Fixed-height segmentation may divide a single table to multiple subpages. Thus we use a dynamic-height pagination strategy: first, the blank lines are detect via the slide window; concretely, if the color deviations of a W x H window is 0, we consider this window is a blank line. The W is set to 1200 to ignore the navigation bar on the left of the pages, and H is set to 10. Then a subset of blank lines are selected such that the width-to-height ration of every segment, excepting the last one, is close to 1: 1.

[6] https://github.com/puppeteer/puppeteer

A.3 Table annotation.

The table annotations are derived from HTML table tags, *e.g.*, ⟨table⟩ translates to a table and ⟨tr⟩ translates to a table row. The pseudo-code to process the HTML table annotation with Puppeteer is summarized as below. Puppeteer API returns valid bounding box for visible elements and returns NULL for invisible ones. The annotation process keeps only visible tables and table elements. Additionally, the merged table cells are annotated with non-empty *row_span* and *col_span* values.

---

function HTML_TO_TABLE(DOM)

```
    Initialize: Annotations = list()
    for T in DOM.elements where T.tag=⟨table⟩ do
        let CurTable=dictionary()
        for R in T.elements where R.tag=⟨tr⟩ do
            let CurRow=dictionary()
            for C in R.elements do
                cell ← annotate_cell(C)
                if cell ≠ Null then
                    CurRow.cells.add(cell)
                end if
            end for
            if c.type=HEADER ∀c ∈ row.cells then
                row.type=HEADER
            end if
            CurTable.rows.add(row)
        end for
        table ← infer_box(CurTable)
        Annotations.add(table)
    end for
    return Annotation
end function
```

---

function ANNOTATE_CELL(C)

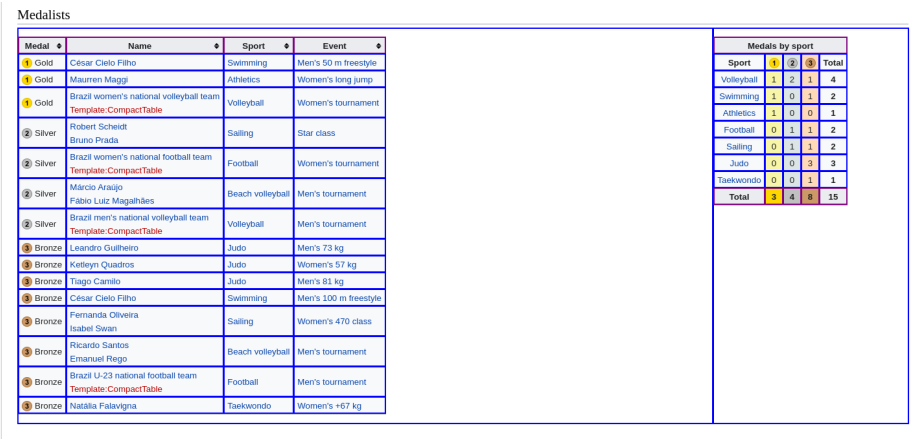
```
    if C is ⟨th⟩ element then
        cell.type = HEADER
    else if C is ⟨td⟩ element then
        cell.type = BODY
    else
        return NULL
    end if
    cell.text ← C.text
    cell.row_span ← C.row_span
    cell.col_span ← C.col_span
    return cell
end function
```

---

function INFER_BOX(T)

$\text{T.box.x0} \leftarrow \min_{R \in T.rows}(\text{R.box.x0})$
$\text{T.box.y0} \leftarrow \min_{R \in T.rows}(\text{R.box.y0})$
$\text{T.box.x1} \leftarrow \max_{R \in T.rows}(\text{R.box.x1})$
$\text{T.box.y1} \leftarrow \max_{R \in T.rows}(\text{R.box.y1})$
end function

Web table filtering. The editors on Wikipedia can use ⟨table⟩ tag to express a broader range of data (*e.g.*, legend as in Figure 11). We remove those false tables by criteria that a true table should have at least two rows and two columns. Moreover, the nested tables are possible in web page while hardly seen in other document format. We notice that some web pages in our dataset contains nested tables (*e.g.*, Figure 7), the outer tables of which are usually used to produce certain layout. Therefore, we detect the nested tables by table bounding boxes and keep only the inner-most tables.



Medalists

Medal	Name	Sport	Event
1 Gold	César Cielo Filho	Swimming	Men's 50 m freestyle
1 Gold	Maurren Maggi	Athletics	Women's long jump
1 Gold	Brazil women's national volleyball team Template:CompactTable	Volleyball	Women's tournament
2 Silver	Robert Scheidt Bruno Prada	Sailing	Star class
2 Silver	Brazil women's national football team Template:CompactTable	Football	Women's tournament
2 Silver	Márcio Araújo Fábio Luiz Magalhães	Beach volleyball	Men's tournament
2 Silver	Brazil men's national volleyball team Template:CompactTable	Volleyball	Men's tournament
3 Bronze	Leandro Guilheiro	Judo	Men's 73 kg
3 Bronze	Ketleyn Quadros	Judo	Women's 57 kg
3 Bronze	Tiago Camilo	Judo	Men's 81 kg
3 Bronze	César Cielo Filho	Swimming	Men's 100 m freestyle
3 Bronze	Fernanda Oliveira Isabel Swan	Sailing	Women's 470 class
3 Bronze	Ricardo Santos Emanuel Rego	Beach volleyball	Men's tournament
3 Bronze	Brazil U-23 national football team Template:CompactTable	Football	Men's tournament
3 Bronze	Natália Falavigna	Taekwondo	Women's +67 kg

Medals by sport

Sport	1	2	3	Total
Volleyball	1	2	1	4
Swimming	1	0	1	2
Athletics	1	0	0	1
Football	0	1	1	2
Sailing	0	1	1	2
Judo	0	0	3	3
Taekwondo	0	0	1	1
Total	3	4	8	15

Fig. 7: Nested tables: the outer tables annotations are removed.

Creating the table extraction task. Figure 8 summarizes the data acquisition and processing of WikiDT table extraction dataset. Following the convention in PubTables-1M, we divide the table extraction to **table detection** which predicts the table bounding boxes from the subpage, and **table structure recognition** which predicts the inner structure in the table from the table crop. The table extraction task data is labeled using Pascal VOC format.

A.4 OCR and Textract table recognition

Realistically, in Document VQA task, ground truth OCR annotation is unavailable. Instead, datasets usually provide the OCR results from off-the-shelf systems. Following the same principle, the OCR and table recognition from a publicly available system is provided along with the images for Document VQA tasks. We used the AWS Textract service to acquire the OCR and table extraction

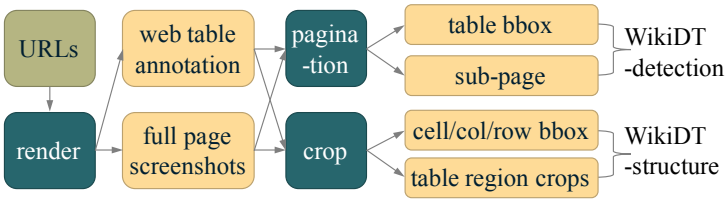


Fig. 8: Data acquisition for WikiDT table extraction task.

(1) page width=1600

(2) page width=500

Fig. 9: Page and table layout in different page width.

results on the subpages. In practice, we also found the Textract results quality on the subpages is substantially better than the quality on full pages. [7] [8]

[7] https://aws.amazon.com/textract/

[8] The annotations are obtained before this update and recognize no merged cell. https://aws.amazon.com/about-aws/whats-new/2022/03/amazon-textract-updates-tables-check-detection/

A.5 QA table processing

Inspected the WikiSQL and WikiTableQuestion samples with the webpage, there are two observations: 1) the tables to the QA pair could not be the information tables, which are shown on the upper right of the page), and could not be reference tables, which are at the end of the page; 2) the tables to the QA pairs can not have only one content row. Therefore, we remove those tables in the QA task.

Furthermore, the content of tables are normalized in the following ways. First, we remove the JavaScript code that are extracted as texts by Puppeteer, which are detected by regular expression matching to keywords (*e.g.*, *mw-parser-output*, *navbar*). Then, if any cell text is longer than 200 characters, it is replaced with a special token *⟨TLDR⟩*. Lastly, the string normalization method adopted in WikiTableQuestions is applied to each table cell text.

Beyond table filtering and table content normalization, the table headers, if contains more than one rows, are flattened. If the table headers are row-headers and multi-level, the headers are transformed using *level1.level2...* format, except for the level where the text for all the cells in that level is the same, *i.e.*, a uniform level value. If the table has only column header, the table are transposed. If the table is 2D, only the row header is considered as header.

A.6 Table retrieval labels.

Table retrieval is an essential step if there are many tables on the page. The table retrieval labels are generated automatically, primarily by retrieving the table with the highest recall using the reference tables and answers from the original dataset (*i.e.*, WikiTableQuestions and WikiSQL) as queries. The process is illustrated in Figure 10. Note that the retrieval label generation process does not provide any cheating method for the table retrieval task: the label generation is based on a reference table, which is not included in the WikiDT dataset and should be assumed unknown to the model, and the answer, which is also unknown to the retrieval model. The table retrieval model of the user of WikiDT should use only the question as a query.

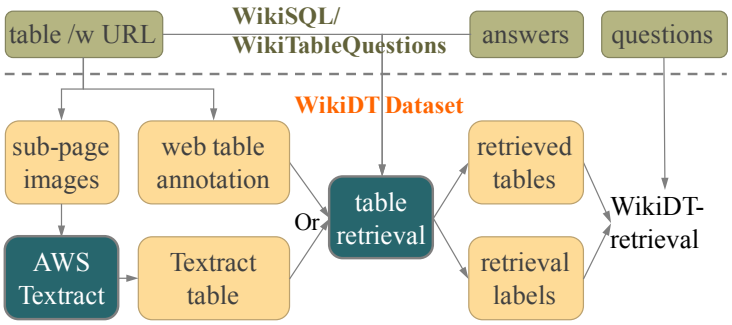


Fig. 10: Data acquisition and processing for WikiDT table retrieval sub-task.

Algorithm 1 shows the table retrieval label generation process. The reference tables and answers from WikiSQL and WikiTableQuestions serve as queries and the candidate tables are regarded as keys. We record the retrieved table by best recall to the reference table and reference keys. In most cases (74%), the two results agree. For the QA experiments, we use the best recall to

the reference table results, since the reference answers may not appear in the table owing to the aggregation functions.

Algorithm 1 Retrieval(reference table $\mathcal{R}$, answers set $\mathcal{A}$, candidate tables $[C_j]_1^n$)

$\mathcal{Q} \leftarrow$ set(x.text for x $\in \mathcal{R}$)
for $C_j \in [C_j]_1^n$ **do**
  $\mathcal{K}_j \leftarrow$ set(x.text for x $\in C_i$)
  Table recall $R_j^{table} \leftarrow \mathcal{K}_j \cap \mathcal{Q}$
  Answer recall $R_j^{answer} \leftarrow \mathcal{K}_j \cap \mathcal{A}$
end for
return $arg\max_j R_j^{table}$, $arg\max_j R_j^{answer}$

Qualified for the quarterfinals

Team v·t·e	Pld	W	D	L	GF	GA	GD	Points
Russia	5	4	1	0	148	125	+23	**9**
South Korea	5	3	1	1	155	127	+28	**7**
Hungary	5	2	1	2	129	142	-13	**5**
Sweden	5	2	0	3	123	137	-14	**4**
Brazil	5	1	1	3	124	137	-13	**3**
Germany	5	1	0	4	123	134	-11	**2**

Fig. 11: False table (upper left one) and true table (lower).

A.7 Answer Generation

Inspecting the table retrieval results from web table annotations, we found that the contents of tables in the WikiSQL dataset are crucially different from the contents on the images, presumably due to the web content updates and post-processing during the creation of the WikiSQL dataset. Therefore, we translated the SQL query from the WikiSQL dataset and executed it on the retrieved web table in WikiDT to generate the answers to the questions, shown in Figure 12. On the other hand, the tables in WikiTableQuestions do not have this issue, thus the answers are still valid given the images and the questions.

B Compare to existing VQA datasets

Table 2 compare the input modality and task formulation of the notable VQA datasets, and a concrete examples of the VQA samples are shown in Figure 13. It can be clearly seen that WikiDT provides a unique type of QA tasks, and are highly challenging in the plenitude of the context information.

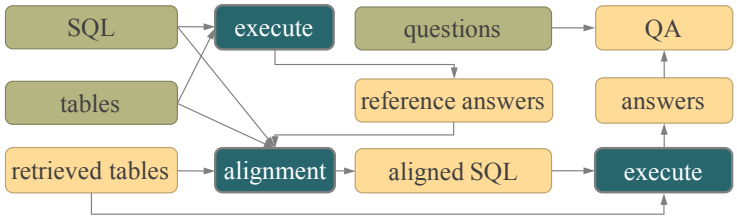


Fig. 12: QA pair generation from WikiSQL dataset from acquired images and tables.

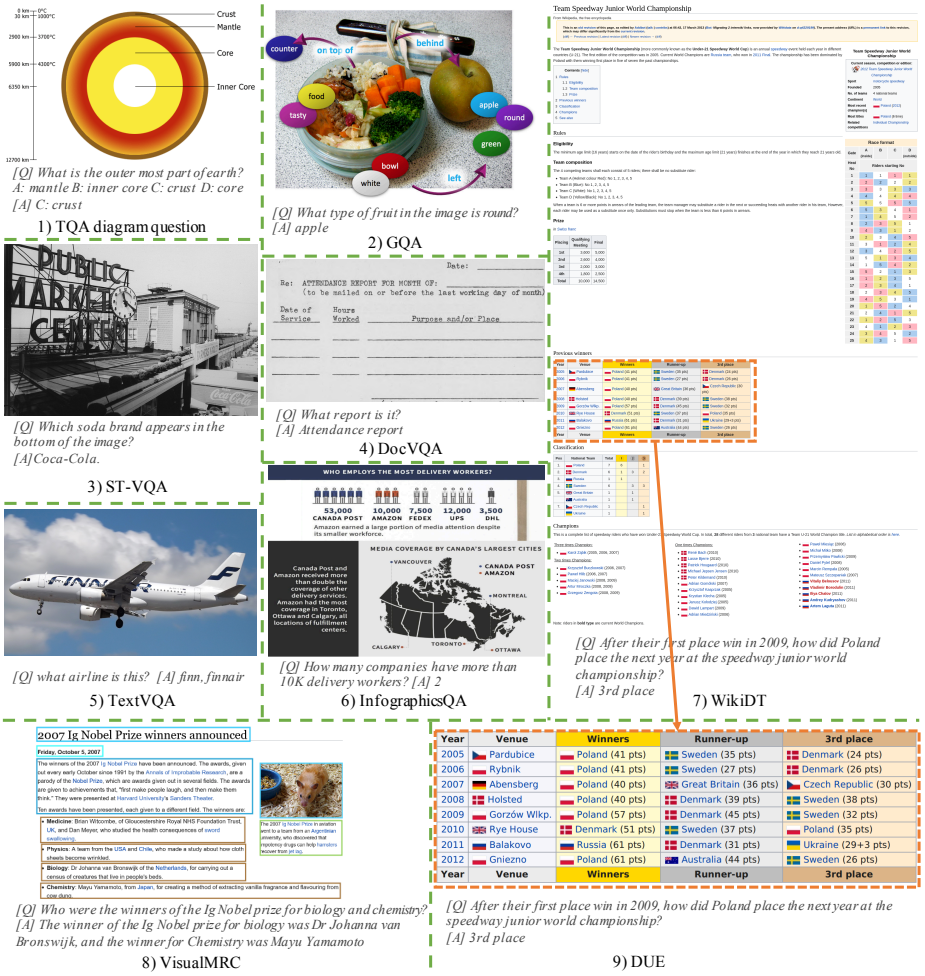


Fig. 13: Illustration of VQA datasets

C Dataset Description and Analysis

C.1 Tasks and datasets.

WikiDT-detection dataset. With the results of pagination, we take the subpage images and the table bounding boxes, ignoring the table structures and content, as a table detection dataset. Images with no tables are discarded.

WikiDT-structure dataset. Independent with the pagination, each table region is cropped from the full-page images to form the image inputs to the table structure recognition task. The coordinates of the bounding boxes are translated from full-page coordinates to the cropped image coordinates. The following types of structural objects are annotated directly from raw web table annotation: table body rows, header rows, table cells, and merged cells. Since the HTML tags do not have table column tags while columns are essential labels for many existing table extraction models (*e.g.*, TableNet[28]), we compute the column bounding boxes and include them in the annotation as well.

WikiDT-TableVQA dataset. The question answering task is to predict the answer to a question given the full page image in an end-to-end manner. Meanwhile, the intermediate labels along the reasoning chains are available for breaking down and diagnosing the models. The intermediate labels include table annotation, table retrieval labels, and SQL queries in most of the samples. The OCR results from Textract are highly accurate and could be leveraged directly to fit into existing document VQA or scene-text VQA frameworks.

C.2 Data analysis.

Images and subpages. Figure 14 shows the image height distribution before (full page) and after (sub-page) pagination step. It shows the pagination greatly reduced average image heights and extremely long pages. The outliers that still have large image heights may contain long tables that the pagination module is unable to split. Figure 15 shows that the number of tables per image reduces after pagination.

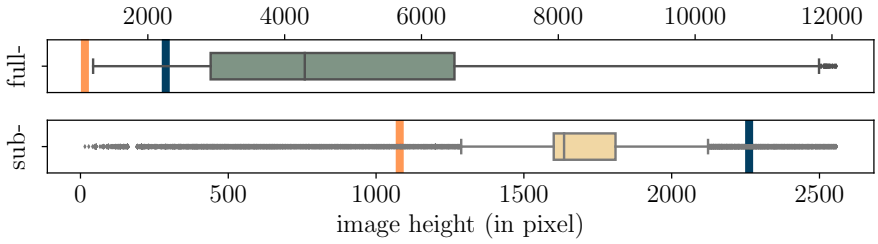


Fig. 14: Image Heights Distributions (x-axis in 10^3).

Page layout. Comparing to existing table recognition dataset created from PDF, the detection task of WikiDT is challenging with diverse table shapes and positions, illustrated in Figure 17. PubTables-1M images are single or double columned PDFs. The table position and width, especially in the double-columned documents are highly similar. On the contrary, table location and shape are much more arbitrary in WikiDT. Figure 16 plots the distribution of table bounding

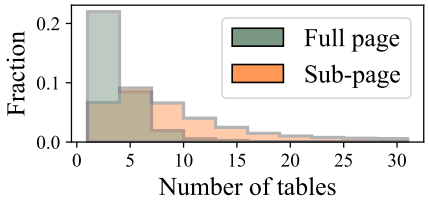


Fig. 15: Number of tables on a single image.

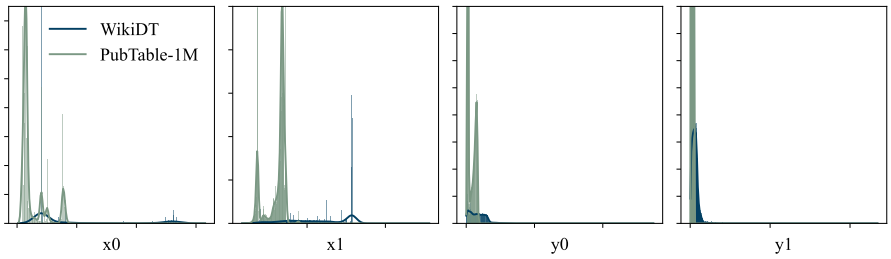


Fig. 16: Table location distribution.

boxes of both dataset. It's obviously seen that the PubTables-1M has few spikes indicating the few locations that are highly likely to be the table border, while the distribution in WikiDT are almost uniform which indicates a diversity in the table location. Moreover, WikiDT contains list-like text blocks that could be easily mis-detected as tables.

Table styles. The table structure recognition task is featured with diversified table styles, shown in Figure 18.

Qualitative annotation results from Textract. AWS Textract gives overall high quality table recognition results, but there are still wrong recognition cases shown in Figure 19. In the experiment section, we compare the two approaches that utilize the Textract Table recognition results with TAPAS model, and another one uses the OCR boxes. Though the TAPAS model has overall better performance, its function relies on the correct recognition results and could not learn to rectify the table recognition error, like the one in Figure 19 (4), when the two columns are recognized as one.

IOU	0.5			0.75			0.95		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
	0.901	0.558	0.689	0.844	0.523	0.646	0.703	0.435	0.538

Table 10: Textract table recognition accuracy.

Top answer items. Table 11 shows the most frequent numeric answers and the text answers. Usually, the numeric answers are produced from some aggregations. For instance, count the number of rows that satisfy certain criteria, or comparing the difference between two values (see

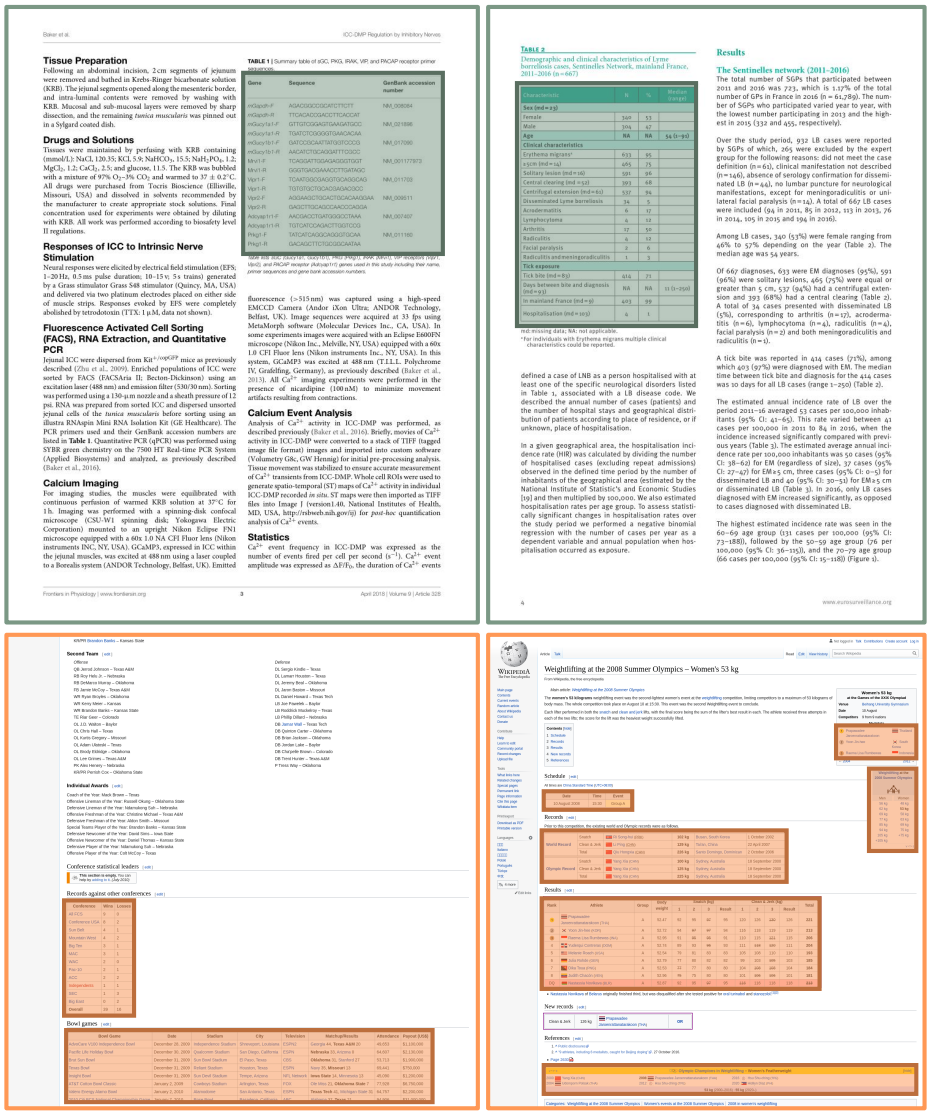
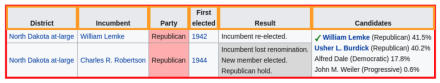


Fig. 17: Table detection samples from PubTables-1M (*upper*) and WikiDT (*lower*).

Figure 22 for example). To this end, understanding the implied reasoning from the questions is critical to solve the QA task.

D Additional Table Recognition Results

Figure 19, 20, 21 illustrate the qualitative table detection and table recognition results. Generally, the DETR model and Textract can generate reliable results, despite that viewing from Table 8 the table extraction performance still have potential to improve.



District	Incumbent	Party	First elected	Result	Candidates
North Dakota at-large	William Lemke	Republican	1942	Incumbent re-elected.	✓ **William Lemke** (Republican) 41.5%
North Dakota at-large	Charles R. Robertson	Republican	1944	Incumbent lost renomination. New member elected. Republican hold.	**Usher L. Burdick** (Republican) 40.2% Alfred Dale (Democratic) 17.8% John M. Weiler (Progressive) 0.6%

(1) First row as header

Processor	Model	Clock speed (GHz)	FSB speed (MT/s)	L2 cache (MB)	CPUs	Cores per CPU	Introduced	Discontinued
Core Duo ("Yonah")	iMac	1.83–2.00	667	2	1	2	January 2006	September 2006
	MacBook Pro	1.83–2.16	667	2	1	2	February 2006	October 2006
	Mac mini	1.66–1.83	667	2	1	2	February 2006	August 2007
	MacBook	1.83–2.00	667	2	1	2	May 2006	November 2006
Core Solo ("Yonah")	Mac mini	1.50	667	2	1	1	February 2006	September 2006
Pentium M ULV ("Crofton")	Apple TV[1][2][3]	1.00	350	2	1	1	January 2007	September 2010

(2) Two dimensional table

Rank	Team	Pts	Matches W	Matches L	Points W	Points L	Points Ratio	Sets W	Sets L	Sets Ratio
1	Kazakhstan	**6**	3	0	266	198	1.343	9	2	4.500
2	Kuwait	**5**	2	1	275	270	1.019	7	6	1.167
3	Myanmar	**4**	1	2	199	233	0.854	4	6	0.667
4	Vietnam	**3**	0	3	236	275	0.858	3	9	0.333

(3) Two-rows as header

Prior to this competition, the existing world and Olympic records were as follows.

World Record	Snatch	Ri Song-hui (PRK)	102 kg	Busan, South Korea	1 October 2002
	Clean & Jerk	Li Ping (CHN)	129 kg	Ta'an, China	22 April 2007
	Total	Qiu Hongxia (CHN)	226 kg	Santo Domingo, Dominican	2 October 2006
Olympic Record	Snatch	Yang Xia (CHN)	100 kg	Sydney, Australia	18 September 2000
	Clean & Jerk	Yang Xia (CHN)	125 kg	Sydney, Australia	18 September 2000
	Total	Yang Xia (CHN)	225 kg	Sydney, Australia	18 September 2000

(4) No header table

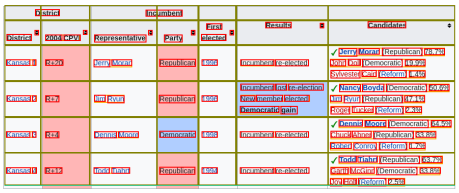
Fig. 18: Various types of table headers.

Numbers		Texts	
Item	Count	Item	Count
'1'	4696	yes	654
'0'	1556	no	512
'2'	1532	united states	398
'3'	1289	incumbent re-elected	158
'4'	1048	canada	132
'5'	900	lr	128
'6'	751	1-1	121
'7'	688	race	116
'8'	540	democratic	111
'9'	473	2010	104

Table 11: Top answer items from WikiDT-TableQA/VQA task.

E TableVQA additional results

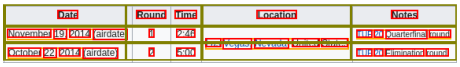
Figure 22 shows when TAPAS makes correct and wrong predictions. Notice that in the wrong prediction case, the question requires a multi-step aggregation. The QA system needs to count the number of ships wrecked in Lake Huron and in Lake Erie separately, then compare the difference. However, by design, TAPAS could solve questions with no more than one aggregation.



(1) Correct recognition

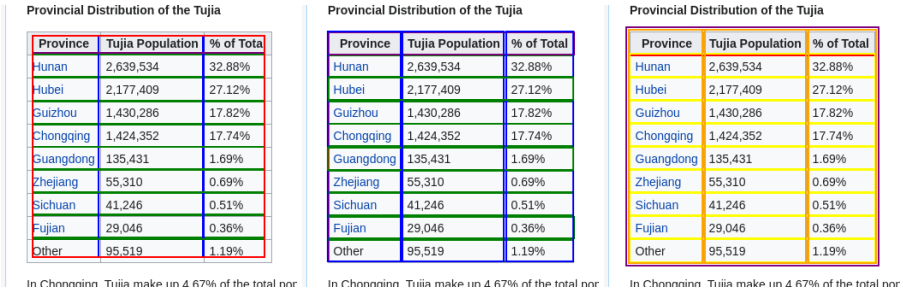
(2) Miss detection

(3) False detection



(4) Merged columns

Fig. 19: Textract recognition results. Red boxes show OCR tokens, olive boxes show table structure.

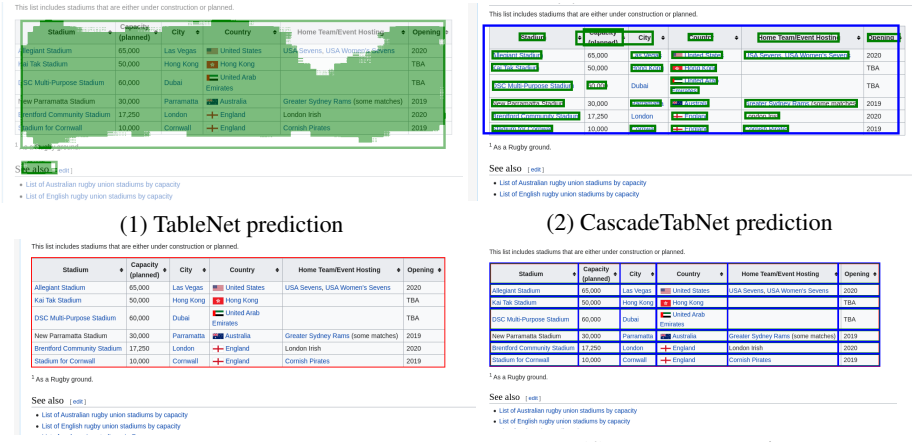


Provincial Distribution of the Tujia

Province	Tujia Population	% of Total
Hunan	2,639,534	32.88%
Hubei	2,177,409	27.12%
Guizhou	1,430,286	17.82%
Chongqing	1,424,352	17.74%
Guangdong	135,431	1.69%
Zhejiang	55,310	0.69%
Sichuan	41,246	0.51%
Fujian	29,046	0.36%
Other	95,519	1.19%

(1) Pretrained DETR (2) DETR (3) ground truth

Fig. 20: DETR prediction compared with ground truth.



(1) TableNet prediction (2) CascadeTabNet prediction

(3) DETR prediction (4) Textract annotation

Fig. 21: Table extraction result comparison.

How many more ships were wrecked in Huron than in Erie?
Predicted aggregator: Count
Final prediction: 7

Ship	Type of Vessel	Lake	Location	Lives lost
Argus[23]	Steamer	Lake Huron	25 miles off Kincardine, Ontario	25 lost
James Carruthers[24]	Steamer	Lake Huron	near Kincardine	18 lost
Hydrus[25]	Steamer	Lake Huron	near Lexington, Michigan	28 lost
Leafield[26]	Steamer	Lake Superior		all hands
John A. McGean[27]	Steamer	Lake Huron	near Goderich, Ontario	28 lost
Plymouth[28]	Barge	Lake Michigan		7 lost
Charles S. Price[29]	Steamer	Lake Huron	near Port Huron, Michigan	28 lost
Regina[30]	Steamer	Lake Huron	near Harbor Beach, Michigan	
Issac M. Scott[31]	Steamer	Lake Huron	near Port Elgin, Ontario	28 lost
Henry B. Smith[28]	Steamer	Lake Superior		all hands
Wexford[27]	Steamer	Lake Huron	north of Grand Bend, Ontario	all hands
Lightship No. 82[28]	Lightship	Lake Erie	Point Albino (near Buffalo)	6 lost

What's the total number of festivals that occurred in October?
Predicted Aggregator: Count
Final prediction: 5

Date	Festival	Location	Awards	Link
Feb 2–5, Feb 11	*Santa Barbara International Film Festival*	Santa Barbara, California USA	*Top 11 "Best of the Fest" Selection*	sbiff.org
May 21–22, Jun 11	*Seattle International Film Festival*	Seattle, Washington USA		siff.net
Sep 28	*Fantastic Fest*	Austin, Texas USA		FantasticFest.com
Oct 9	*London Int. Festival of Science Fiction Film*	London, England UK	*Closing Night Film*	Sci-Fi London
Oct 9, Oct 11	*Sitges Film Festival*	Sitges, Catalonia Spain		Sitges Festival
Oct 1, Oct 15	*Gwacheon International SF Festival*	Gwacheon, Gyeonggi-do South Korea		gisf.org
Oct 17, Oct 20	*Icon TLV*	Tel Aviv, Central Israel		icon.org.il
Oct 23	*Toronto After Dark*	Toronto, Ontario Canada	Best Special Effects Best Musical Score	torontoafterdark.com
Nov 11	*Les Utopiales*	Nantes, Pays de la Loire France		utopiales.org

Fig. 22: TAPAS prediction examples (left: correct, right: wrong). Light blocks show the column selection result and the darker blocks show the cell selection.