ICDAR 2023 Competition on Visual Question Answering on Business Document Images

Sachin Raja, Ajoy Mondal(✉), and C. V. Jawahar

International Institute of Information Technology, Hyderabad, India
sachin.raja@research.iiit.ac.in, {ajoy.mondal,jawahar}@iiit.ac.in

Abstract. This paper presents the competition report on Visual Question Answering (VQA) on Business Document Images (VQAonBD) held at the 17th International Conference on Document Analysis and Recognition (ICDAR 2023). Understanding business documents is a crucial step toward making an important financial decision. It remains a manual process in most industrial applications. Given the requirement for a large-scale solution to this problem, it has recently seen a surge in interest from the document image research community. Credit underwriters and business analysts often look for answers to a particular set of questions to reach a decisive conclusion. This competition is designed to encourage research in this broader area to find answers to questions with minimal human supervision. Some problem-specific challenges include an accurate understanding of the questions/queries, figuring out cross-document questions and answers, the automatic building of domain-specific ontology, accurate syntactic parsing, calculating aggregates for complex queries, and so on. Further, despite having the same accounting fundamentals, the terminologies and ontologies used across different organizations and geographic locations may vary significantly. This makes the problem of generic VQA on such documents only more challenging. Since this is the first iteration of the competition, it was restricted in terms of some of the challenges listed; however, the further iterations of this competition aim to include many additional sub-tasks with the larger vision of accurate semantic understanding of business documents as images.
Eleven different teams around the world registered for this competition. Five teams out of those submitted methods spanning multiple approaches, among which Team Upstage KR won the competition with a weighted average score of 95.9%. The runner-up team, NII-TablQA obtained a weighted average score of 90.1%

Keywords: Optical Character Recognition (OCR) · Visual Question Answering (VQA) · Business Documents · Table Structure Recognition (TSR)

1 Introduction

Visual question-answering generally aims to answer a query described in natural language, taking cues from the document image as the only input. As a part of

G. A. Fink et al. (Eds.): ICDAR 2023, LNCS 14188, pp. 454–470, 2023.
https://doi.org/10.1007/978-3-031-41679-8_26

this competition, we propose a visual question-answering dataset and baseline model from business document images. While a lot of work has already been done in the broader VQA space [1–11], the questions from business documents present many niche challenges that may require cross-document referencing, additional numeric computations over the simple search query to reach the final solution, and so on. Further, since most business documents are usually presented in a tabular format, leveraging this structural conformity to answer more challenging queries may be non-trivial. Given the unique nature of the problem, its tremendous prospect in the industry, layers of challenges to be tackled, and the recent surge of interest in visual question answering, we believe that there would be a surge in the research interest in this area in the near future (Fig. 1).

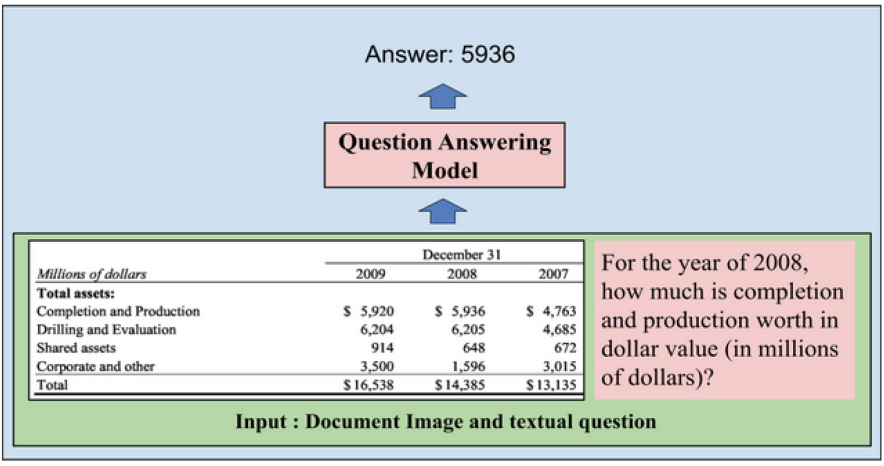

Fig. 1. Given a document image and questions, the task of the competition is to produce answers corresponding to the questions.

The recent works in the broader problem of visual question answering on generic scene images demonstrate the ability of deep-learning models to understand the context of the scene at hand. While at first glance, the problem of document VQA, particularly VQA on tabular images seems quite similar, the reality is quite different. Tabular data often presents highly dense data compressed in a structured format, with limited linguistic contexts. This is usually because most of the data present is in numeric format, which is more complex to digitize and understand in the broader context than standard documents containing sentences and paragraphs.

Another possible way to approach the problem may be through a more pipeline-driven methodology which would need table detection and table reconstruction as precursors. Though this process involves multiple stages, it would result in easy explainability with respect to question-answeing. Moreover, the success of recent methods in table reconstruction space [12–19] make this approach as a reasonable prospective.

The problem at hand has an immense utility, primarily in banking and insurance verticals where analysts manually digitize the incoming financial reports

(including but not limited to balance sheets, income statements and cash flow statements). As a next step, subject matter experts, such as credit underwriters, peruse these reports to extract answers for a specific set of queries to make a decision. This competition aims to pose this problem as a cognitive machine learning task to answer the queries at hand, given only the table image along with queries as inputs.

The paper is organized as follows. In Sect. 2, we give details about the dataset used for the competition. The submitted methods are discussed in Sect. 3. Section 4 shows the results of the competition. The conclusive remark is drawn in Sect. 5.

2 Dataset

We use the publicly available FinTabNet [20] datasets for this competition. FinTabNet [20] dataset has predefined ground truth labels for table structure recognition, which means that alongside every image, we have bounding boxes for every word/token, digitized text, and row/column identifier. We create questions on top of these documents and tag their answers in terms of the actual textual answer by annotating the word/token bounding box(es) used to compute the final answer. Some of the complex table images with multiple row and column headers split across different columns and rows respectively are shown in Figs. 2, 3 and 4.

In order to achieve the desired scale of the dataset, we employ a heuristic-based automated algorithm to create the dataset using original table structure annotations of the FinTabNet [20] dataset. The algorithm, in brief, is described as follows:

- As a first step, we get the table grid from original FinTabNet [20] annotations that allow us to identify all table cells, including those which are empty.
- The next step is to identify the data-type of each cell based on its content. The data-types include string, integer, floating-point, empty, percentage-value, year, month, date, special chars and ranges, to name a few.
- Once the datatype of every cell is identified, we employ heuristics to identify row headers and column headers depending on data-types and whether the cell spans multiple rows and/or columns.
- In case the image contains multiple tables, as shown in Figs. 2, 3 and 4, we use the header information to split the tables horizontally and/or vertically.
- At this point, all the information and metadata (row headers, column headers, cell data-types) of the table are extracted.
- Most business report document tables can often be represented in a tree-like structure where certain rows add up corresponding to a row below in the table in a recursive manner. We extract this tree structure for every table in the dataset to identify inter-row relationships.
- Lastly, we generate questions of varying difficulty levels using all the table-level and cell-level metadata collected as described above.

	Amount of Gain / (Loss) Recognized in OCI				Amount of Gain / (Loss) Reclassified from OCI		
	Year Ended December 31,				Year Ended December 31,		
Derivative Instrument	2013	2012	2011	Location on Statement of Earnings	2013	2012	2011
Foreign exchange forward contracts	$63.9	$16.3	$(34.9)	Cost of products sold	$ 8.0	$(12.0)	$(32.9)
Foreign exchange options	(0.3)	(1.1)	(0.2)	Cost of products sold	(0.2)	(0.4)	–
Cross-currency interest rate swaps	–	–	0.2	Interest expense	–	0.2	(8.3)
	$63.6	$15.2	$(34.9)		$ 7.8	$(12.2)	$(41.2)

Fig. 2. Example of a complex table image that has row headers split across different columns.

The different categories of questions imply varying difficulty levels of the questions as described below:

(In millions)	Zions Bank			Amegy			CB&T		
	2017	2016	2015	2017	2016	2015	2017	2016	2015
SELECTED INCOME STATEMENT DATA									
Net interest income	$ 650	$ 624	$ 544	$ 483	$ 460	$ 387	$ 476	$ 434	$ 377
Provision for loan losses	19	(22)	(28)	25	163	91	(5)	(9)	(4)
Net interest income after provision for loan losses	631	646	572	458	297	296	481	443	381
Noninterest income	151	149	133	118	123	121	75	67	63
Noninterest expense	436	424	430	336	326	373	299	290	294
Income before income taxes	$ 346	$ 371	$ 275	$ 240	$ 94	$ 44	$ 257	$ 220	$ 150
SELECTED AVERAGE BALANCE SHEET DATA									
Total average loans	$ 12,481	$ 12,538	$ 12,118	$ 11,021	$ 10,595	$ 10,148	$ 9,539	$ 9,211	$ 8,556
Total average deposits	15,986	15,991	15,688	11,096	11,130	11,495	11,030	10,827	10,063

(In millions)	NBAZ			NSB			Vectra		
	2017	2016	2015	2017	2016	2015	2017	2016	2015
SELECTED INCOME STATEMENT DATA									
Net interest income	$ 206	$ 190	$ 152	$ 134	$ 122	$ 94	$ 126	$ 120	$ 101
Provision for loan losses	(8)	(3)	8	(11)	(28)	(28)	1	(8)	5
Net interest income after provision for loan losses	214	193	144	145	150	122	125	128	96
Noninterest income	40	40	36	40	39	36	25	23	21
Noninterest expense	148	144	133	139	137	131	101	97	98
Income before income taxes	$ 106	$ 89	$ 47	$ 46	$ 52	$ 27	$ 49	$ 54	$ 19
SELECTED AVERAGE BALANCE SHEET DATA									
Total average loans	$ 4,267	$ 4,086	$ 3,811	$ 2,357	$ 2,284	$ 2,344	$ 2,644	$ 2,469	$ 2,400
Total average deposits	4,762	4,576	4,311	4,254	4,137	3,891	2,756	2,720	2,792

(In millions)	TCBW			Other			Consolidated Company		
	2017	2016	2015	2017	2016	2015	2017	2016	2015
SELECTED INCOME STATEMENT DATA									
Net interest income	$ 46	$ 38	$ 28	$ (56)	$ (121)	$ 32	$ 2,065	$ 1,867	$ 1,715
Provision for loan losses	2	—	(3)	1	—	(1)	24	93	40
Net interest income after provision for loan losses	44	38	31	(57)	(121)	33	2,041	1,774	1,675
Noninterest income	5	5	4	90	70	(57)	544	516	357
Noninterest expense	20	19	17	170	148	105	1,649	1,585	1,581
Income (loss) before income taxes	$ 29	$ 24	$ 18	$ (137)	$ (199)	$ (129)	$ 936	$ 705	$ 451
SELECTED AVERAGE BALANCE SHEET DATA									
Total average loans	$ 926	$ 791	$ 707	$ 266	$ 88	$ 87	$ 43,501	$ 42,062	$ 40,171
Total average deposits	1,107	1,007	879	1,209	207	(481)	52,200	50,595	48,638

Fig. 3. Example of a complex table image that have column headers split across different rows.

- To generate category 1 questions, which are simple extraction queries, we define multiple question templates and depending on the cell data-type and metadata, we curate the question accordingly.
- For the questions of category type 2, we compute ratios of cells that belong to the same row but across two different columns. The question is then curated according to the pre-defined multi-paraphrased templates by populating the corresponding values of the row header and the two-column headers.
- For the questions of category type 3, we compute ratios of cells across two different rows. The question is then curated according to the pre-defined multi-paraphrased templates by populating the corresponding values of the row and column headers.
- For the questions of category type 4, we compute aggregation functions (among minimum, maximum, mean, median and cumulative) across cells with the same row header but belonging to different years or months of the report. The question is then curated according to the pre-defined multi-paraphrased templates by populating the corresponding values of the row and column headers (years).
- For the questions of category type 5, we make use of the recursive inter-rows relationships to compute aggregation (among minimum, maximum, mean, median and cumulative) across a group. The questions around these groups are generated from the same column header and group row headers of the report. The question is then curated according to the pre-defined multi-paraphrased templates by populating the corresponding values of the row and column headers.

(In thousands)	Net unrealized gains (losses) on investment securities	Net unrealized gains (losses) on derivatives and other	Pension and post-retirement	Total
2015				
Balance at December 31, 2014	$ (91,921)	$ 2,226	$ (38,346)	$ (128,041)
Other comprehensive income (loss) before reclassifications, net of tax	(12,471)	4,903	(3,161)	(10,729)
Amounts reclassified from AOCI, net of tax	86,023	(5,583)	3,718	84,158
Other comprehensive income (loss)	73,552	(680)	557	73,429
Balance at December 31, 2015	$ (18,369)	$ 1,546	$ (37,789)	$ (54,612)
Income tax expense (benefit) included in other comprehensive income (loss)	$ 48,422	$ (331)	$ 374	$ 48,465
2014				
Balance at December 31, 2013	$ (168,805)	$ 1,556	$ (24,852)	$ (192,101)
Other comprehensive income (loss) before reclassifications, net of tax	82,204	2,275	(15,284)	69,195
Amounts reclassified from AOCI, net of tax	(5,320)	(1,605)	1,790	(5,135)
Other comprehensive income (loss)	76,884	670	(13,494)	64,060
Balance at December 31, 2014	$ (91,921)	$ 2,226	$ (38,346)	$ (128,041)
Income tax expense (benefit) included in other comprehensive income (loss)	$ 60,795	$ 467	$ (8,764)	$ 52,498

Fig. 4. Example of a complex table image that have column headers (year of the table) split across different rows.

During the training phase, the dataset is divided into two categories - training and validation sets containing 39,999 and 4535 table images respectively. Ground truth corresponding to each table image consists of the following: Table Structure Annotation: Each cell is annotated with information about its bounding box, digitised content, and cell spans in terms of start-row, start-column, end-row and end-column indices. Difficulty-Wise Sample Questions and Answers: Corresponding to every table image, a few sets of questions along with their answers are annotated in the JSON file. The questions are organised into five categories in increasing order of difficulty. The question types primarily include extraction type query, ratio calculations and aggregations across rows and/or columns. Further, answer types are classified as text or numeric. While text answers will be evaluated according to edit-distance-based measures, for numeric-type answers, the absolute difference between the ground-truth and predicted value will also be taken into account. Ideally, to answer all the questions correctly, both syntactic along with a semantic understanding of the business document would be required. Each table image would have annotations for a maximum of 50 questions and corresponding answers for training and validation. Depending on the format and content of the table, the total number of questions from each category within a single table will be in the following range:

- Category 1 : 0–25
- Category 2 : 0–10
- Category 3 : 0–3
- Category 4 : 0–7
- Category 5 : 0–5

Every training annotation is in the form of a json file that contains two primary keys:

- Table Structure (table_structure): Each key within this object is represented by an integer value, cell_id. The object corresponding to this cell_id has information about its bounding box, start_row, start_col, end_row, end_col and content.
- Questions and Answers (questions_answers): The keys within this object denote the category of questions (category_1, etc.). Further, the object corresponding to each category is again a dictionary with a key corresponding to the question_id and a value corresponding to the question_object containing the question as the string, its answer and answer type.

During the evaluation, the predictions are expected in a similar JSON format such that the key at the first level is the category_id. Within each category is a nested dictionary such that its key is the question_id and the corresponding value is the predicted answer.

The statistics of the dataset are as shown below:

Table 1. Division of dataset into training, validation, and test sets. **#:** indicates counts.

Dataset-Type	#Images	#Total Questions	#Numeric Questions	#Text Questions
Training	39,999	1,254,165	1,197,358	56,807
Validation	4,535	141,465	134,651	6,814
Test	4,361	135,825	129,861	5,964

3 Methods

In this section, we discuss each of the submitted methods including the baseline in detail. Eleven teams registered for the competition. However, we obtained complete submissions from five of them, which include results, submission reports, trained model(s) and inference codes. One team did submit the results but did not submit other details to test for reproducibility and hence, won't be included in the leaderboard. These five final participants are:

Table 2. Category-wise distribution of questions in training, validation and test datasets. **#:** indicates counts.

Question	Training Dataset		Validation Dataset		Test Dataset	
Category	#Numeric	#Text	#Numeric	#Text	#Numeric	#Text
Category 1	632,037	56,807	69,458	6,814	68,439	5,964
Category 2	137,395	0	15,396	0	14,705	0
Category 3	107,712	0	12,471	0	11,863	0
Category 4	187,844	0	21,696	0	20,609	0
Category 5	132,370	0	15,630	0	14,245	0

- **Upstage KR**, affiliation: Upstage
- **NII-TABIQA**, affiliation: National Institute of Informatics, Japan
- **DEEPSE-X-UPSTAGE-HK**, affiliation: DeepSE x Upstage HK
- **BD-VQA**, affiliation: Apple Inc.
- **SFANC57**, affiliation: OneConnect FinTech

3.1 Baseline

We evaluated the method proposed by Xu and Li. [21,22] for our baseline. In their work, they proposed the model called **LayoutLM**, which jointly models interactions between text and layout information across scanned document images. This becomes beneficial for a great number of real-world document image understanding tasks such as information extraction from scanned documents.

To add to this, authors also leverage image features to incorporate words' visual information into LayoutLM. Their architecture extends the well-known Bert [23] model by adding two types of input embeddings: (i)a 2-D position embedding that denotes the relative position of a token within a document; and (ii) an image embedding for scanned token images within a document. The proposed 2-D position embedding captures the relationship among tokens within a document, meanwhile, the image embedding captures visual characteristics including but not limited to fonts, font-styles, colors, etc. In addition, authors employ a multi-task learning objective for LayoutLM [21,22], which includes a Masked Visual-Language Model (MVLM) loss and a Multi-label Document Classification (MDC) loss. The two losses combined allow for joint pre-training of text and layout collectively. It is important to note that in order to extract the token, authors use an OCR tool as a precursor to the joint training. The pre-trained model was then finetuned on form understanding, receipt understanding and document image classification as the downstream tasks. The implementation that we have employed is in the form of an API available on Hugging-Face, which has been further finetuned on both the SQuAD2.0 [24] and DocVQA [6] datasets. This makes it a go to choice for our baseline[1]

3.2 Upstage KR

Participants use three models named CPRQ (Component Prediction from Raw Question), CPEQ (Component Prediction from Extracted Questions), and CPEQ Pseudo. After the prediction of each model, they generate the final result using weighted hard voting (Table 1).

CPRQ. Component Prediction from Raw Question (CPRQ) attempts to train the generative model (Donut [25]) to predict the values of the components needed to answer the original raw question. Taking the ratio-type questions as an example, instead of training the model to predict the final ratio answer, it was trained to output the values present within the table that are needed to solve the ratio question. After successfully extracting of the necessary component values, subsequent mathematical operations (e.g. ratio) could be applied in the post-processing step. To obtain the component values corresponding to the different mathematical operation questions, both rule-based algorithms and external generative model API were used. For the external generative model API, ChatGPT 3.5 [26] API to be specific, only the training dataset was used to find the component values and train their model (Table 2).

CPEQ. Component Prediction from Extracted Questions (CPEQ) attempts to train the generative model (Donut [25]) to predict component value from extracted questions.

[1] The code is available at https://huggingface.co/impira/layoutlm-document-qa.

First, a raw question is divided into multiple extractive questions similar to those in category 1 by pre-defined rules. For example, "What is the ratio of the value of due after 10 years for the year 2018 to the year 2017?" is divided into two extractive questions such as "What is the ratio of the value of due after 10 years for the year 2018?" and "What is the ratio of the value of due after 10 years for the year 2017?". Participants defined some dividing patterns that can cover all questions.

Second, a trained model using only category 1 data as training data predict both category 1 questions and extracted questions in categories 2–5. Lastly, predictions from extracted questions in categories 2–5 are post-processed to generate the final result by operation (e.g. maximum, minimum, ratio).

CPEQ Pseudo. CPEQ is trained using only category 1 data. For data augmentation, pseudo question-answer pair is generated by the CPEQ algorithm, and the trained CPEQ model is fine-tuned on pseudo data. The resulting model is CPEQ Pseudo.

3.3 NII-TablQA

The team introduces TabIQA, a system designed for question-answering using table images in business documents, as illustrated in Fig. 5. Given a table image of a business document and a question about the image, the system utilizes the table recognition module to extract table structure information and the text content of each table cell and convert them into HTML format. Subsequently, the high-level table structure is extracted to identify the headers, data cells, and hierarchical structure with the post-structure extraction module. Once the table is structured, it is converted to a data frame for further processing. The question-answering module processes the input question and the table data frame with an encoder and generates the final answer from a decoder.

Table Recognition. This module aims to predict the table structure information and the text content of each table cell from a table image and represent them in a machine-readable format (HTML). Specifically, this module consists of one shared encoder, one shared decoder, and three separate decoders for three sub-tasks of table recognition: table structure recognition, cell detection, and cell-content recognition. Participants trained this model on the training set of VQAonBD 2023 and validated it on the validation set of VQAonBD 2023 for model selection and choosing the hyperparameters.

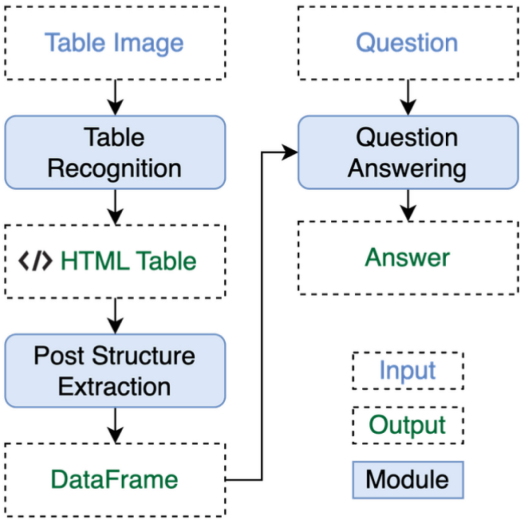

Fig. 5. Architectural diagram of the team NII-TablQA.

Post-structure Extraction. The TabIQA system classifies table headers and data rows from HTML tables using a set of heuristics. Specifically, the system identifies headers as some of the first table rows with column spans, nan cells, or duplicate values in the same rows. The system designates the first row as the table header if no header is found. The system then classifies the remaining rows as data cells. The system identifies hierarchical rows by focusing on data cells with column spans for entire rows. Once the system has identified the structured table, it generates a table data frame by concatenating the values of the header rows to form a one-row header and concatenating the value of each hierarchical row to the lower-level cell values to improve the interpretation of each cell value and provide a more accurate representation of the table data in the data frame.

Question Answering. This module is built on the state-of-the-art table-based question-answering model, OmniTab [27]. The team fine-tuned the OmniTab [27] large pre-trained models using the VQAonBD 2023 training set.

3.4 DeepSE-x-Upstage-HK

Their method, Donut-EAMA (Extract Answer Merge Answer), is based on the end-to-end OCR-free document understanding model - Donut [25] (https://github.com/clovaai/donut). To apply it on the VQAonBD task, they first pre-trained the model on the training set with the text-reading task. Then considering the model had no training involving arithmetic calculations, they believed

that asking it to answer the questions directly would probably not work well. Therefore, the team developed a rule-based algorithm that extracts relevant cell values based on the question and the provided table annotations for the training set and uses those extracted values as labels to reformulate the task into an extractive one. They then finetune the Donut [25] model on this extractive task and implemented a simple post-processing algorithm to calculate the final answer from the values generated by the model.

3.5 BD-VQA

As part of this challenge, the team has used Donut [25] VQA (Visual Question Answering) pre-trained model open-sourced by Hugging face (https://huggingface.co/naver-clova-ix/donut-base). This model is a deep learning model that is designed to answer questions about images of donuts.

Before feeding the image and question list as inputs into the Donut VQA system [25], they performed data pre-processing, handling questions from different categories in distinct ways. They left Category 1 questions as they were, while for Category 2 and Category 3 questions, they split them into two independent questions and subsequently computed the ratio of the two values in the table. This was done because they noticed that these questions relied on the ratio of two values.

For Category 4 and 5 questions that involved operations such as median, maximum, minimum, cumulative, and average, were found to rely on the final aggregate output of three values in the table. Hence, the team split them into three separate questions. Using Donut VQA [25], they predicted the value of each question, and then computed the corresponding operator value to obtain the final result.

3.6 SFANC57

For the system used for VQAonBD, the team has chosen the OCR-free VDU model Donut [25]. For category 1 questions: most answers can be directly selected from the original table content; thus we generate the answer from the Donut-VQA model. For category 2–5 questions, they developed a simple query parsing script to split the logic into content selection and aggregation calculation.

4 Evaluation

4.1 Evaluation Metrics

During the evaluation, a model is expected to take only the document image and question as the input to produce the output. This output is then compared against the ground truth answer to obtain a quantitative evaluation score computed over the entire evaluation dataset.

In most cases, the expected answers to questions from business documents are single numeric token ones. It makes classical accuracy a good prospect for

evaluating this task. While for a more generic assignment of visual question answering, there may be some subjectivity in the answers (e.g., white, off-white, and cream may all be correct answers), the solutions for the proposed task are primarily objective and absolute. It makes evaluation relatively straightforward. Hence, we use standard accuracy as the primary criterion for evaluation. Further, we also employ averaged absolute deviation as one of the criteria for numeric-type answers. If the absolute difference between the ground truth and the predicted value is more than 100%, we give a score of 0. In the other case, the score is defined by:

$$Deviation\ Score\ =\ 1\ -\ \frac{absolute\ distance}{ground\ truth\ value} \tag{1}$$

However, since the input to the model will only be by the document image to answer a specific query, penalizing the VQA model word/token detection and recognition is not fair. Therefore, we also employ Averaged Normalized Levenshtein Similarity (ANLS) as proposed in [28,29], which responds softly to answer mismatches due to OCR imperfections. ANLS is given by Eq. 2, where N is the total number of questions, M are possible ground truth answers per question, $i\ =\ 0...N$, $j\ =\ 0...M$ and o_{q_i} is the answer to the i^{th} question q_i.

$$\begin{aligned} ANLS &= \frac{1}{N}\sum_{i=0}^{N}\left(\max_{j}\ s(a_{ij}, o_{q_i})\right) \\ s(a_{ij}, o_{q_i}) &= \begin{cases} 1 - NL(a_{ij}, o_{q_i}), & \text{if } NL(a_{ij}, o_{q_i}) < \tau. \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{2}$$

where $NL(a_{ij}, o_{q_i})$ is the normalized Levenshtein distance (ranges between 0 and 1) between the strings a_{ij} and o_{q_i}. The value of τ can be set to add softness toward recognition errors. If the normalized edit distance exceeds τ, it is assumed that the error is because of an incorrectly located answer rather than an OCR mistake.

The final score is an L2 norm of the deviation score and the ANLS score, both of which range between 0 and 1 for the numeric values. For text answers, the final score is the same as the ANLS score.

4.2 Results

Out of eleven registered participants, we received submissions from a total of six teams. Five of them submitted their results along with a brief description of their method, trained model(s) and inference codes. The final leaderboard consists of those five submissions. Furthermore, we have executed the inference codes for each of the submissions to ensure that the submission score could be replicated within a $+-1\%$ score. Also, we received multiple submissions from each team. To ensure there was no cherry-picking of the best-performing submission, we only considered the most recent submission by the team within the deadline window.

Table 3. Final Scores corresponding to the latest submissions of all the participating teams. Categories 1 through 5 indicate the average scores corresponding to questions of each category, All Avg indicates the average scores and Weighted average indicates the weighted average score, based on which the final ranking was decided.

Team	Category 1	Category 2	Category 3	Category 4	Category 5	All Avg	Weighted Avg
Baseline	0.281	0.091	0.096	0.200	0.169	0.168	0.163
UPSTAGE KR	0.963	0.942	0.953	0.974	0.956	0.957	0.959
NII-TABIQA	0.932	0.876	0.855	0.895	0.931	0.898	0.901
DEEPSE-X-UPSTAGE-HK	0.939	0.874	0.859	0.902	0.858	0.886	0.879
BD-VQA	0.799	0.794	0.729	0.736	0.422	0.696	0.640
SFANC57	0.648	0.119	0.132	0.463	0.418	0.356	0.359

Table 4. Final exact match accuracy scores corresponding to the latest submissions of all the participating teams. Categories 1 through 5 indicate the average exact match scores corresponding to questions of each category, All Avg indicates the average exact match scores and Weighted average indicates the weighted average exact match score.

Team	Category 1	Category 2	Category 3	Category 4	Category 5	All Avg	Weighted Avg
BASELINE	0.085	0.000	0.000	0.015	0.012	0.023	0.015
UPSTAGE KR	0.933	0.907	0.925	0.957	0.924	0.929	0.931
DEEPSE-X-UPSTAGE-HK	0.872	0.799	0.784	0.791	0.734	0.796	0.778
BD-VQA	0.586	0.630	0.533	0.501	0.110	0.472	0.397
NII-TABIQA	0.874	0.554	0.451	0.215	0.259	0.470	0.374
SFANC57	0.111	0.001	0.002	0.090	0.140	0.069	0.082

Table 5. Evaluation based on answer data types.

Team Team	Numeric Score	Text Score	Micro-Average Score	Numeric Exact Match Score	Text Exact Match Score	Micro Average Exact Match Score
BASELINE	0.214	0.359	0.220	0.051	0.020	0.050
UPSTAGE	0.962	0.929	0.960	0.934	0.880	0.932
NII-TABIQA	0.924	0.674	0.913	0.645	0.470	0.637
DEEPSE-X-UPSTAGE-HK	0.912	0.870	0.910	0.833	0.750	0.829
BD-VQA	0.753	0.522	0.743	0.545	0.051	0.523
SFANC57	0.494	0.470	0.493	0.091	0.057	0.089

From Tables 3 and 4, it is evident that the team **UPSTAGE KR** won the competition by a significant margin of 5.8% average weighted final score across all the categories of questions as compared to the runner-up team, which obtained a score of 90.1%. There are many interesting conclusions that can be drawn from these results. If we only consider the simple extractive questions, which belong to category 1, we observe that the results obtained by the top three teams are within a close range of 3% scores. Among the participants, we observe three very distinct approaches toward the solution. The first team follows a weighted

ensemble-driven approach where they train three different generative models using the architecture of Donut [25] and ChatGPT 3.5 [26] API to answer the questions. The second team, on the other hand, follows a more pipeline-driven approach where they perform table recognition as a precursor step for post-structure data extraction using heuristics to extract row and column headers. On top of the structured information extracted, they use the OmniTab [27] model to generate answers. The third team used the Donut [25] model but reformulated the task into an extractive task instead of a text reading task. The fourth and fifth teams used Donut [25] model to extract answers to the questions. The fourth team developed parsers to break down complex questions into simple ones, while the fifth standing team did not fine-tune or developed any query parsers but used the standard Donut [25] model API available on hugging-face to generate answers.

The numbers clearly indicate that the fine-tuning of the pre-trained generative models like Donut [25] is imperative to obtain any meaningful results in the first place because of completely different dataset distributions. The difference between the scores of the third and fourth teams also clearly indicates the significance of training a problem-specific downstream task for a generative model instead of using it right out of the box. Further, a difference of almost 19% score between the BD-VQA and SFANC57 teams indicates that developing complex question parsers and transforming those into simple extractive queries can significantly aid generative models; however, such models fail to perform well directly on the aggregation and ratio-type complex questions.

Further, Table 5 compares the performance of each submission on text and numeric-type questions. The non-trivial difference between the proposed evaluation score and exact match accuracy scores clearly demonstrates that there is some error induced because of OCR mistakes. The difference however is particularly stark for the team NII-TABIQA. Our qualitative analysis suggests that the difference is primarily in the least significant bits of the numeric values. The significant difference for the same submission for text-based questions further signifies that OCR does not seem to be as accurate as compared to the other submissions.

As discussed above, we draw many interesting conclusions from various submissions of this competition. In this first iteration of the competition, we only requested for the answers of every question put forward in front of the model and did not ask for where the relevant information was picked up from in order to answer the query. This makes it hard for us to thoroughly investigate the errors made by the OCR tool in extracting tokens. In the next version, we would definitely ask for the coordinates of the relevant tokens which would allow us to thoroughly investigate the submissions from the OCR dimension as well.

5 Conclusion

This competition aims to bridge the gap between the document research community in the academia and the industry. Through this competition, we have seen two primary distinct ways in which researchers go about tackling this problem -

(i) through direct VQA on images as a black box; and (ii) a more pipeline-driven approach using table structure recognition and OCR as precursors to answering the query. The high-performing quantitative results show both approaches as promising directions of research in this space.

Since this was the first version of this competition and in turn the dataset, the questions were generated primarily using keywords from the underlying ground-truth tokens of the document itself. Furthermore, the aggregation queries by themselves contained many cues using which it was not so difficult to break them down into simpler questions to answer (as we have seen in most of the submissions). The reasonable number of participants and submissions in this challenge motivates us to take this further and build upon the dataset to make it all the more challenging. Some of the ways in which we plan to do this are to (i) increase the scope of the documents (including invoices, receipts, etc.); (ii) add cross-document questions; (iii) add additional sub-tasks (such as table-specific tokens detection and recognition, table structure recognition, key-value pair detection); and (iv) by building domain specific taxonomy and ontology which would make the questions independent of the absolute keywords seen in the document thereby making them generic for multiple similar style of documents. We believe that in the future, our competition would play a vital role in getting towards a rather "Grand Challenge" in the document research space at large.

In conclusion, we hope that this competition would continue to bridge the gap between the document research community in academia and the industry. We also hope that models presented in this competition will eventually lead to the building of state-of-the-art artificially intelligent methods that could solve the real-world problem efficiently at a large scale.

Acknowledgement. This work is supported by MeitY, Government of India.

References

1. Antol, S., et al.: VQA: visual question answering. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2425–2433 (2015)
2. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 618–626 (2017)
3. Anderson, P., et al.: Bottom-up and top-down attention for image captioning and visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6077–6086 (2018)
4. Zhou, L., Palangi, H., Zhang, L., Houdong, H., Corso, J., Gao, J.: Unified vision-language pre-training for image captioning and VQA. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 13041–13049 (2020)
5. Changpinyo, S., Kukliansky, D., Szpektor, I., Chen, X., Ding, N., Soricut, R.: All you may need for VQA are image captions. arXiv preprint: arXiv:2205.01883 (2022)
6. Mathew, M., Karatzas, D., Jawahar, C.V.: DocVQA: a dataset for VQA on document images. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 2200–2209 (2021)

7. Mishra, A., Shekhar, S., Singh, A.K., Chakraborty, A.: OCR-VQA: visual question answering by reading text in images. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 947–952. IEEE (2019)
8. Yusuf, A.A., Chong, F., Xianling, M.: An analysis of graph convolutional networks and recent datasets for visual question answering. Artif. Intell. Rev. **55**, 1–24 (2022)
9. Lu, J., Yang, J., Batra, D., Parikh, D.: Hierarchical question-image co-attention for visual question answering. In: Advances in Neural Information Processing Systems, vol. 29 (2016)
10. Jang, Y., Song, Y., Yu, Y., Kim, Y., Kim, G.: TGIF-QA: toward Spatio-temporal reasoning in visual question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2758–2766 (2017)
11. Gokhale, T., Banerjee, P., Baral, C., Yang, Y.: VQA-LOL: visual question answering under the lens of logic. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12366, pp. 379–396. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58589-1_23
12. Tensmeyer, C., Morariu, V.I., Price, B., Cohen, S., Martinez, T.: Deep splitting and merging for table structure decomposition. In: ICDAR (2019)
13. Qasim, S.R., Mahmood, H., Shafait, F.: Rethinking table parsing using graph neural networks. In: ICDAR (2019)
14. Qiao, L., et al.: LGPMA: complicated table structure recognition with local and global pyramid mask alignment. In: Lladós, J., Lopresti, D., Uchida, S. (eds.) ICDAR 2021. LNCS, vol. 12821, pp. 99–114. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86549-8_7
15. Zhang, Z., Zhang, J., Jun, D., Wang, F.: Split, embed and merge: an accurate table structure recognizer. Pattern Recogn. **126**, 108565 (2022)
16. Lin, W., et al.: TSRFormer: table structure recognition with transformers. arXiv preprint: arXiv:2208.04921 (2022)
17. Long, R., et al.: Parsing table structures in the wild. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 944–952 (2021)
18. Raja, S., Mondal, A., Jawahar, C.V.: Table structure recognition using top-down and bottom-up cues. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12373, pp. 70–86. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58604-1_5
19. Raja, S., Mondal, A., Jawahar, C.V.: Visual understanding of complex table structures from document images. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 2299–2308 (2022)
20. Zheng, X., Burdick, D., Popa, L., Zhong, X., Wang, N.X.R.: 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, pp. 697–706 (2021)
21. Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., Zhou, M.: 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, pp. 1192–1200 (2020)
22. Xu, Y., et al.: LayoutLMv2: multi-modal pre-training for visually-rich document understanding. arXiv preprint: arXiv:2012.14740 (2020)
23. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint: arXiv:1810.04805 (2018)
24. Rajpurkar, P., Jia, R., Liang, P.: Know what you don't know: unanswerable questions for SQuAD. arXiv preprint: arXiv:1806.03822 (2018)

25. Kim, G., et al.: OCR-free document understanding transformer. In: Avidan, S., Brostow, G., Cisse, M., Farinella, G.M., Hassner, T. (eds.) Computer Vision - ECCV 2022. ECCV 2022. Lecture Notes in Computer Science, vol. 13688, pp. 498–517. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-19815-1_29
26. Hagendorff, T., Fabi, S., Kosinski, M.: Machine intuition: uncovering human-like intuitive decision-making in GPT-3.5. arXiv preprint: arXiv:2212.05206 (2022)
27. Jiang, Z., Mao, Y., He, P., Neubig, G., Chen, W.: OmniTab: pretraining with natural and synthetic data for few-shot table-based question answering. arXiv preprint: arXiv:2207.03637 (2022)
28. Biten, A. F., et al.: ICDAR 2019 competition on scene text visual question answering. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 1563–1570. IEEE (2019)
29. Tito, R., Mathew, M., Jawahar, C.V., Valveny, E., Karatzas, D.: ICDAR 2021 competition on document visual question answering. In: Lladós, J., Lopresti, D., Uchida, S. (eds.) ICDAR 2021. LNCS, vol. 12824, pp. 635–649. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86337-1_42