Table Question Answering for Low-resourced Indic Languages

Vaishali Pal[1,2] **Evangelos Kanoulas**[1] **Andrew Yates**[1] **Maarten de Rijke**[1]
[1]University of Amsterdam, The Netherlands
[2]Discovery Lab, Elsevier, The Netherlands
v.pal, e.kanoulas, a.c.yates, m.derijke@uva.nl

Abstract

TableQA is the task of answering questions over tables of structured information, returning individual cells or tables as output. TableQA research has focused primarily on high-resource languages, leaving medium- and low-resource languages with little progress due to scarcity of annotated data and neural models. We address this gap by introducing a fully automatic large-scale table question answering (tableQA) data generation process for low-resource languages with limited budget. We incorporate our data generation method on two Indic languages, Bengali and Hindi, which have no tableQA datasets or models. TableQA models trained on our large-scale datasets outperform state-of-the-art LLMs. We further study the trained models on different aspects, including mathematical reasoning capabilities and zero-shot cross-lingual transfer. Our work is the first on low-resource tableQA focusing on scalable data generation and evaluation procedures. Our proposed data generation method can be applied to any low-resource language with a web presence. We release datasets, models, and code.[1]

1 Introduction

Tables are ubiquitous for storing information across domains and data sources such as relational databases, web articles, Wikipedia pages, etc. (Deldjoo et al., 2021). Tables introduce new challenges in machine comprehension not present in text as they are are not well-formed sentences but a semi-structured collection of facts (numbers, long-tail named entities, etc.) (Iyyer et al., 2017; Jauhar et al., 2016; Jin et al., 2022; Katsis et al., 2022; Liu et al., 2021; Nan et al., 2022; Pal et al., 2022; Zhu et al., 2021). Additionally, tables make position (rows/columns) bias (Lin et al., 2023) and entity popularity bias (Gupta et al., 2023) severe. The tableQA task introduces novel challenges compared to text-based question answering (textQA) (Herzig et al., 2020; Liu et al., 2021; Ye et al., 2023; Yu et al., 2018; Zhao et al., 2022). In addition to the semi-structured nature of tables, a tabular context leads to a high frequency of fact-based questions, mathematical and logical operations such as arithmetic (Zhu et al., 2021), set, relational (Jiang et al., 2022; Liu et al., 2021), and table operations such as table joins (Pal et al., 2023). Effective tableQA systems not only have machine comprehension skills, but also numeracy understanding (Cheng et al., 2022; Liu et al., 2021; Zhao et al., 2022; Zhu et al., 2021), table reasoning (Liu et al., 2021; Yu et al., 2018), table summarization (Zhang et al., 2024; Zhao et al., 2023a) and answer table generation ability (Pal et al., 2023).

Low-resource tableQA aims to answer questions over semi-structured tables storing cultural and region-specific facts in a low-resource language. Joshi et al. (2020) show that most languages struggle to be represented and are deprived of advances in NLP research. As manual data collection is slow and expensive, low-resource languages struggle with large-scale, annotated data for effective transfer learning solutions. The low-resource setting (Hedderich et al., 2021; Ruder, 2019) exacerbates the challenges of tableQA with challenges of data sparsity, annotated data costs, and lack of trained models. In contrast to textQA, syntactico-semantic variations such as agreement and morphology are not exhibited in tables, but high presence of culturally significant yet long-tail entities makes adapting existing high resource datasets and trained models challenging. Research on low-resource table inference (Minhas et al., 2022) shows that standard approaches of translating English datasets for low-resource data creation are infeasible for tables due to high translation error as tables are not well-formed sentences.

Challenges. Our work focuses on studying the following core challenges of low-resource tableQA:

[1]https://github.com/kolk/Low-Resource-TableQA-Indic-languages

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 75–92
November 12-16, 2024

(1) low-resource **tableQA data scarcity** and under-representation of cultural facts.
(2) Existing **neural models' poor alignment** in low-resource languages and a lack of understanding of table structure.

This motivates us to explore low-resource tableQA by designing a low-cost and large-scale automatic data generation and quality estimation pipeline. We discuss the process in detail with a low-resource Indic language, Bengali (spoken extensively in Bangladesh and India, with over 230 million native speakers (Karim et al., 2021)), and explore generalizability with Hindi (570 million speakers).

Our main contributions are as follows:
(1) We introduce **low-resource tableQA task**.
(2) We design a **method** for automatically generating low-resource tableQA data in a scalable budget-constrained manner.
(3) We release **resources** to support low-resource tableQA: Large-scale tableQA **datasets** and **models** for 2 Indic languages, Bengali (Bengali Table Question Answering (BanglaTabQA)) and Hindi (Hindi Table Question Answering (HindiTabQA)). BanglaTabQA contains $19K$ Wikipedia tables, $2M$ training, $2K$ validation and 165 test samples. HindiTabQA contains $2K$ Wikipedia tables, $643K$ training, 645 validation and 125 test samples.

2 Related Work

TableQA aims to answer a user question from semi-structured input tables. Prior work on tableQA in English can be classified as extractive (Herzig et al., 2020; Yin et al., 2020) or abstractive (Nan et al., 2022; Pal et al., 2022; Ye et al., 2023; Zhao et al., 2023b). While extractive tableQA focuses on row and cell selection (Herzig et al., 2020), abstractive tableQA generates various types of answers such as factoid answers (Liu et al., 2021), summaries (Zhang et al., 2024; Zhao et al., 2023b), or answer tables (Pal et al., 2023). Low-resource setting poses challenges for various NLP tasks. The low-resource corpus creation (Bhattacharjee et al., 2022; Das and Saha, 2022; Hasan et al., 2020) has used automatic annotation efforts by synthesizing a large-scale dataset. Das and Saha (2022) train a Bengali QA system by developing a synthetic dataset translated from standard English QA datasets. Bhattacharjee et al. (2022); Hasan et al. (2020) create low-resource datasets by translating English datasets to Bengali using neural models. However, these methods are unsuitable due to the semi-structured ungrammatical sequential representation of tables.

3 Task Definition

We formulate low-resource tableQA as a sequence generation task. Given a question Q of k tokens $q_1, q_2, \ldots, q_k$, and table T comprising of m rows and n columns $\{h_1, \ldots, h_n, t_{1,1}, t_{1,2}, \ldots, t_{1,n}, \ldots, t_{m,1}, t_{m,2}, \ldots, t_{m,n}\}$ where $t_{i,j}$ is value of the cell at the i-th row and j-th column and h_j is the j-th column header; the low-resource tableQA model generates an answer table T_{out}. The input sequence is the concatenated question Q, and linearized input table T separated by special sentinel tokens. The answer, T_{out}, is also a linearized sequence. Henceforth, for concreteness, we will use Bengali as the example low-resource language. The input to such a model is:

q_1 $q_2 \ldots q_k$ <কলাম> $h_i \ldots h_n$ <রো ১> $t_{1,1} \ldots t_{1,n}$ <রো i> $t_{i,j} \ldots t_{i,n} \ldots$ <রো m> $t_{m,1} \ldots t_{m,n}$.

The answer table, T_{out}, is a linearized sequence:

<কলাম> $H_i \ldots H_q$ <রো ১> $o_{1,1} \ldots o_{1,q}$ <রো i> $o_{i,j} \ldots o_{i,q} \ldots$ <রো m> $o_{p,1} \ldots o_{p,q}$

where $o_{i,j}$ is value at the i-th row and j-th column and H_j is the j-th column header of T_{out}.

4 Methodology for Dataset Generation

Effective training of low-resourced tableQA requires creation of large-scale datasets of questions, input and answers tables, to align a language model to the low-resource language and adapt it to semi-structured tables and QA task. We address **Challenge 1** by designing an automatic data generation process to generate a large-scale low resource tableQA corpus of training and validation samples. We follow a 3-step pipeline as follows: (i) table extraction, (ii) question generation, and (iii) answer table extraction. This pipeline applied on Bengali, as depicted in Figure 1, generates the **BanglaTabQA** dataset.

4.1 Table Extraction

English Wikipedia with $6,751,000+$ articles is used for English tableQA datasets (Pasupat and Liang, 2015), but is insufficient for non-Latin languages with many cultural topics missing. The standard process (Bhattacharjee et al., 2022; Das and Saha, 2022) of translating English datasets to low-resource languages is biased due to lack of cultural topic/fact representation in English tableQA

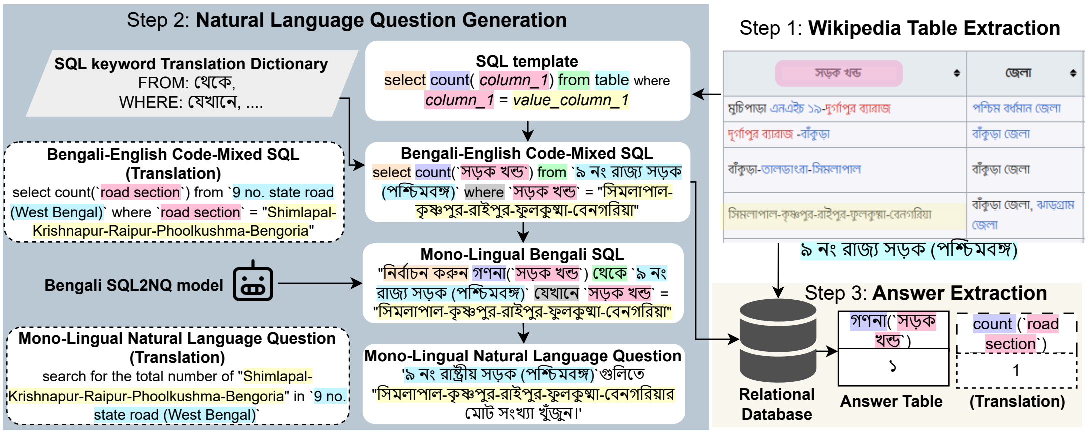

Figure 1: **BanglaTabQA Dataset generation**: The SQL elements and table elements are color-coordinated to represent a single SQL/table element. Dotted rectangles represent translations for accessibility to non-native readers.

datasets. For example, the named-entity অধিরাজ গাঙ্গুলি (Adhiraj Ganguly), exists only in Bengali Wikipedia,[2] and not in English. Further, translating English tables with machine translation models is error-prone (Minhas et al., 2022) as tables are not well-formed sentences but collections of facts. To mitigate these issues, we extract tables from Wikipedia dump of the low-resource language.

4.2 Natural Language Question Generation

The question generation is a 2-step process:

Code-mixed SQL query generation. We automatically generate SQL queries over the extracted low-resourced tables with SQL templates from the SQUALL dataset (Shi et al., 2020). These templates have placeholders of table components such as table name, column names, etc. which are randomly assigned with values from a Wikipedia table. For example, the template "`select count(c1) from w where c1 = value`" is instantiated by assigning a Bengali table name "৯ নং রাজ্য সড়ক (পশ্চিম বঙ্গ)" to `w`, column header "জেলা" to `c1`, and "বাঁকুড়া জেলা" to `value`. This results in an executable code-mixed query "`select count(জেলা) from ৯ নং রাজ্য সড়ক (পশ্চিম বঙ্গ) where 'জেলা' = "বাঁকুড়া জেলা"`", where the SQL keywords are in English but all table information is in the low-resource language (Bengali). This leads to $13,345,000$ executable Bengali code-mixed queries.

Natural language question generation. We formulate question generation as a sequence-to-sequence task by transforming a code-mixed SQL query into a natural language question (NQ). To the best of our knowledge, there exists no sequence generation models which translates code-mixed SQL queries to low-resource natural language questions. To train a model for this conversion, we first transform the code-mixed SQL to a monolingual SQL-like query in the low-resource language. As the only linguistic variation exhibited in the SQL templates is polysemy i.e. a dearth of one-to-one correspondence between English SQL keywords and the corresponding low-resource language translations, we employ native speakers well-versed in SQL to manually create one-to-one mappings of 27 SQL keywords for linguistic transfer of SQL keywords to the corresponding low-resource language. All table-specific words are directly copied into the monolingual query. We discard `FROM` keyword and table name from the query as it is associated with a single input table. This leads to a SQL-like monolingual query in the low-resource language which is a well-formed sentence. For example, code-mixed Bengali query "`select count( 'জেলা') from ৯ নং রাজ্য সড়ক (পশ্চিম বঙ্গ) where 'জেলা' = "বাঁকুড়া জেলা"`", results in a monolingual Bengali query "`নির্বাচন করুন গণনা( 'জেলা') যেখানে 'জেলা' = "বাঁকুড়া জেলা"`". In contrast to tables which are invalid sentences, queries and NQ are well-formed sequences and effectively transformed (SQL to question) with existing encoder-decoder models. We train a SQL-to-NQ (SQL2NQ) model (`mbart-50-large` (Liu et al., 2020) backbone) by translating $68,512$ training and $9,996$ validation samples from semantic parsing datasets: Spider (Yu et al., 2018), WikiSQL (Zhong et al., 2017), Atis (Dahl et al., 1994; Price, 1990), and Geoquery (Zelle and Mooney, 1996) to the low-resource language. We use this SQL2NQ model to transform the queries to NQ. For example, Bengali SQL2NQ model transforms the aforementioned query to the NQ "কবার বাঁকুড়া জেলার উল্লেক আছে?".

[2] https://bn.wikipedia.org/wiki/অধিরাজ_গাঙ্গুলি

4.3 Answer Table Extraction

We dump low-resource Wikipedia tables in a relation database. The code-mixed SQL queries are executed with an SQL compiler over a relational database comprising of the low-resourced Wikipedia tables to extract the answer tables. We execute the $13,345,000$ Bengali code-mixed queries to extract the corresponding answer tables.

4.4 Automatic Quality Control

We employ automatic quality control steps to ensure quality of the synthetic tableQA data.

Code-mixed query and answer quality control. We discard all code-mixed queries which execute to an error with an SQL compiler. This process follows the quality control in (Pal et al., 2023) and discards invalid and erroneous queries and samples.

Natural Language Question quality control. We evaluate the quality of the generated NQ with a sentence similarity model to discard questions that have low similarity score with the corresponding monolingual queries. We found the standard method of quality evaluation in low-resource languages (Bhattacharjee et al., 2022; Ramesh et al., 2022) using the sentence similarity model, LaBse (Feng et al., 2022), incompatible for code-mixed SQL-NQ due to low discriminating ability (0.55 mean similarity score and 0.13 standard deviation for Bengali SQL-NQ). For example, LaBse assigns low score (0.43) for positive SQL-NQ pair corresponding to the Bengali query “`SELECT title ORDER BY year DESC LIMIT 1`" and Bengali NQ “`Return the most recent title corresponding to the most recent year`" (translated for non-native readers), while it assigns a high score (0.8) to negative pair “`SELECT count(*) WHERE 'work' = The World of Saudamini`" and the unrelated NQ “`How many games scored a total of 4?`". Table 10 in Appendix A.8 shows more examples. This necessitates fine-tuning LaBse on low-resourced SQL-NQ samples. First, we use the translated semantic parsing samples ($68,512$ training and $9,996$ SQL-NQ pairs), described in Section 4.2, as positive pairs and in-batch negatives with multiple-negatives ranking loss. We call this the SQL2NQSim model. We select the best checkpoint by evaluating SQL2NQSim on $1,000$ randomly selected hard-negatives (unrelated/negative SQL-negative question pairs for which pre-trained LaBse assigns a high similarity score (> 0.5)). We use that checkpoint to obtain similarity scores of the low-resourced tableQA SQL-NQ pairs and discard samples with a similarity score lower than a threshold. We select a good threshold by plotting a histogram of scores assigned by the SQL2NQSim model on $10,000$ randomly selected positives and hard-negatives and selecting the inflection point as the threshold. Figure 2 shows the scores' histogram for BanglaTabQA. We select a strict threshold of 0.74 (hard-negatives scores taper-off around 0.7). The final BanglaTabQA dataset, after quality control, comprises of $2,050,296$ training and $2,053$ validation samples.

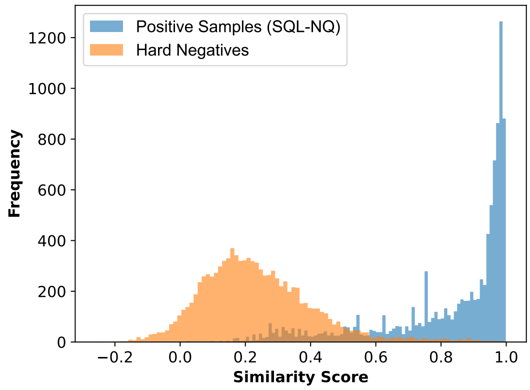

Figure 2: Histogram of similarity scores from fine-tuned Bengali SQL2NQSim model of $1,000$ random samples

4.5 Dataset Analysis

In contrast to textQA, tableQA focuses on mathematical questions (Liu et al., 2021; Pal et al., 2023; Zhu et al., 2021). Following (Liu et al., 2021), we analyse BanglaTabQA dataset on question complexity, which estimates the difficulty of a question based on the corresponding SQL query. As tableQA enforces mathematical, logical and table reasoning questions, we further classify tableQA queries into different classes of table operations determined by the SQL operators present.

Question complexity. Recent work on tableQA (Liu et al., 2021) categorizes SQL queries into difficulty levels based on the number of SQL keywords. We follow this approach and count the number of keywords for each query. Figure 3 shows that most of BanglaTabQA queries have 4 SQL keywords. The longest SQL queries are comprised of 10 keywords, and the shortest ones of 3 SQL keywords.

Mathematical operations. We further categorize each sample based on the operators present in

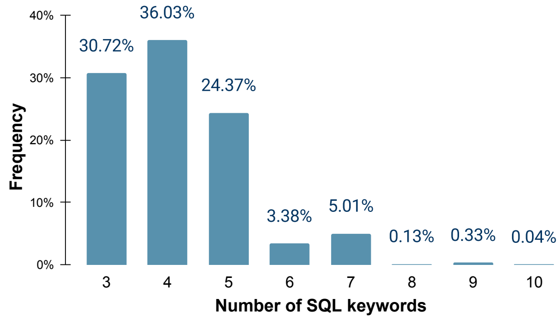

Figure 3: Number of SQL keywords per query histogram in the BanglaTabQA dataset.

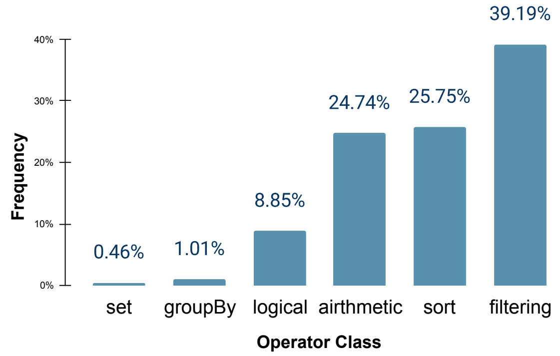

Figure 4: Histogram of operator classes in the BanglaTabQA dataset.

the question. We utilize the SQL query associated with a question to extract all keywords for classification. We categorize data samples into 6 operator classes: arithmetic, sorting, group by, filtering, set operators, and logical operators. Arithmetic operators comprises of SQL numeric operations such as `sum`, `count`, `min`, etc. Sorting refers to ordering of the answer values in an ascending or descending order. Group by is the SQL operator of grouping rows based on a criterion. Filtering corresponds to SQL operators such as `where` and `having` used to filter the input table. Set operators involve `union`, `intersect`, and `except`. Finally, we classify logical operators to be conjunction (`and`) and disjunction (`or`) to combine filtering conditions. It also includes membership operators (`in`, `between`, etc.) and string matching operator (`like`). The classification of the operators is shown in Table 3. Figure 4 shows the distribution of the 6 operator classes for the BanglaTabQA dataset.

4.6 Test Set

We manually annotate test samples for evaluating low-resource tableQA models on clean data. We select unique tables not present in the training and validation set to avoid data leakage. To ensure question diversity, we select code-mixed SQL representing each of the 6 operator classes (discussed in Section 4.5) and distinct from the training and validation data. Three native annotators well-versed in SQL were employed for annotation. One annotator was tasked with question generation and given the synthetic SQL query, input tables and the answer table, and asked to rewrite the code-mixed query to a natural language question. The remaining two were tasked with evaluation of the question generated by the first annotator. The evaluator-annotators were provided the code-mixed query, input table, answer table, and the annotated question and asked to rate the question based on fluency. We estimate the annotated question fluency with a 5-point Likert scale (1-5), where a higher score indicates a better fluency. The final score for each question was computed by averaging the scores of the evaluator-annotators. For BanglaTabQA, we manually annotate 165 test samples. We estimate an inter-annotator agreement with Fliess's Kappa score (Fleiss, 1971) of 0.82, indicating strong agreement among the annotators. The average fluency score across test set questions was 4.3, indicating high fluency.

4.7 Generalizability of Dataset Methodology

We study the generalizability of the dataset generation method by repeating the process on another Indic language: Hindi (Hi) with more than 602 million speakers. To the best of our knowledge, there is no existing tableQA data for Indic languages. Hindi text is in Devanagari script which is different from Bengali written in Eastern-Nagari (Bengali-Assamese) script. This requires tableQA models to be trained on large-scale Hindi datasets for good alignment. Following the dataset creation process in Section 4, we extract $1,921$ Hindi tables from the respective Wikipedia dumps. We generate $82,00,000$ Hindi code-mixed queries automatically to extract answer tables and generate the Hindi natural language questions. The final HindiTabQA dataset comprises of $643,434$ synthetic training, 645 synthetic validation samples and 121 manually annotated test samples.

5 Experimental Setup

We address **Challenge 2** by studying the effectiveness of state-of-the-art models (baselines) in *Bengali table QA*. Experimental results (Section 6) show the need for a large-scale BanglaTabQA dataset and model training. We analyze several models' effectiveness in Bengali language, mathematical/table operations and generalizability, thus providing a measure of the dataset quality and consequently the dataset creation methodology.

Baselines. We perform 2-shot in-context learning (ICL) to adapt large language model (LLM)s to BanglaTabQA task. We further fine-tune an encoder-decoder model. The demonstrations are the concatenated question and flattened input table with the flattened answer table. We use the following models as baselines:

(1) **En2Bn:** We fine-tune an encoder-decoder model, `mbart-50-large`, with $25,000$ random samples from MultiTabQA's (Pal et al., 2023) pre-training data translated to Bengali using Google translate. MultiTabQA used SQUALL templates to generate their queries and have the same distribution as BanglaTabQA queries. However, the input tables of MultiTabQA are English wiki-tables from WikiTableQuestions dataset (Pasupat and Liang, 2015) and are not representative of Bengali cultural topics/facts.

(2) **OdiaG** (**Parida et al., 2023**) is Llama-7b (Touvron et al., 2023) adapter-tuned (LoRA (Hu et al., 2022)) on 252k Bengali instruction set.[3]

(3) **GPT:** `GPT-3.5` (Brown et al., 2020) performs well on English tableQA (Zha et al., 2023). `GPT-4` (OpenAI et al., 2023) outperforms other LLMs (Chinchilla (Hoffmann et al., 2022), PaLM (Chowdhery et al., 2022)) in low-resource languages, including Bengali and Hindi, on various tasks ($14,000$ multiple-choice problems on 57 subjects in a translated MMLU benchmark (Hendrycks et al., 2021)).

BanglaTabQA models. Bengali tableQA models must understand both Bengali **script** *and* **numerals**, crucial for mathematical operations. However, Bengali numbers are not present in many state-of-the-art Indic models' (Dabre et al., 2022; Gala et al., 2023)[4] vocabulary. To the best of our knowledge, there is no open-access generative model which understands both table structure and Bengali. We train the following models on BanglaTabQA as they support Bengali and Hindi numbers and text:

(1) **BnTQA-mBart:** `mbart-50-large` (Liu et al., 2020) is a multi-lingual encoder-decoder model with support for 50 languages.

(2) **BnTQA-M2M:** `m2m100_418M` (Fan et al., 2021) is a multi-lingual encoder-decoder model with support for 100 languages.

(3) **BnTQA-llama:** We train `Llama-7B`, on BanglaTabQA dataset with parameter-efficient fine-tuning (PEFT) on LoRA adapters.

We train `BnTQA-mBart` and `BnTQA-M2M` with 128 batch size and `BnTQA-llama` with 16 batch size and 4-bit quantization. All models are trained with 1e-4 learning rate on a single A6000 48GB GPU for 5 epochs with 1024 maximum sequence length.

5.1 HindiTabQA

We assess the generalizabiltiy of our data generation process by training and evaluating HindiTabQA models. All hyper-parameters and experimental setup are the same as Bengali.

Baselines. We use the following baselines:

(1) **En2Hi:** Similar to `En2Bn`, we fine-tune `mbart-50-large` with $25,000$ random samples from MultiTabQA, translated to Hindi.

(2) **GPT**: We perform 2-shot ICL on the best LLMs on Bengali, `GPT-3.5` and `GPT-4`.

(3) **OpHathi**: We perform 2-shot ICL on `OpenHathi-7B-Hi-v0.1-Base`, an open-source LLM based on `llama-7b` and trained on Hindi, English, and Hinglish text.

HindiTabQA models. We train the following models on the HindiTabQA dataset:

(1) **HiTQA-llama:** Similar to Bengali, we fine-tune `Llama-7b` on HindiTabQA dataset.

(2) **HiTQA-M2M:** Similar to Bengali, we fine-tune `m2m100_418M` on HindiTabQA dataset.

(3) **HiTQA-mBart:** Similar to Bengali, we fine-tune `mbart-50-large`, on HindiTabQA.

(4) **HiTQA-BnTQA:** `BnTQA-mBart`, trained on BanglaTabQA provides a warm start. We fine-tune it on HindiTabQA for better convergence.

5.2 Evaluation Metrics

The answer table requires both table structure and content evaluation rendering standard text similarity metrics (Rouge, BLEU, etc.) inappropriate. We instead evaluate with tableQA evaluation metrics (Pal et al., 2023). Henceforth, F1 scores are the harmonic mean of the precision and recall scores.

(1) **Table Exact Match Accuracy (Tab)** measures the percentage of generated answer which *match exactly* to the target answer tables.

(2) **Row Exact Match F1 (Row)**: Row EM precision is the percentage of correctly predicted rows among all predicted rows. Row EM recall

[3]OdiaGenAI/odiagenAI-bengali-lora-model-v1
[4]ai4bharat/IndicBART

Model	Bengali								Hindi							
	Validation Set scores (%)				Test Set scores (%)				Validation Set scores (%)				Test Set scores (%)			
	Tab	**Row**	**Col**	**Cell**	**Tab**	**Row**	**Col**	**Cell**	**Tab**	**Row**	**Col**	**Cell**	**Tab**	**Row**	**Col**	**Cell**
En2(Bn/Hi)	0.05	3.06	0.20	3.07	0.00	4.73	0.00	4.73	0.00	3.37	0.47	3.43	0.00	5.03	8.26	5.03
OdiaG	0.00	3.89	0.00	3.89	0.69	1.77	0.69	1.42	–	–	–	–	–	–	–	–
OpHathi	–	–	–	–	–	–	–	–	0.00	0.00	0.00	0.00	0.00	0.11	0.37	0.74
GPT-3.5	1.14	4.81	1.67	5.14	6.04	10.06	9.12	9.84	4.81	8.94	4.99	9.71	8.20	10.29	7.10	9.81
GPT-4	0.00	13.57	5.43	14.65	26.83	**38.67**	26.74	**36.51**	15.53	22.60	16.02	22.25	11.11	21.49	11.76	20.84
	BnTQA								**HiTQA**							
-llama	**60.08**	**68.30**	**60.47**	**68.30**	9.41	12.35	9.85	11.87	14.76	9.92	14.13	7.29	13.11	9.71	11.11	7.66
-mBart	56.63	64.10	56.79	64.31	**35.88**	33.16	**35.88**	33.16	92.09	87.97	92.02	87.97	33.06	43.35	33.88	43.35
-M2M	45.31	58.07	45.29	58.04	28.05	34.55	28.05	34.55	89.55	85.32	89.34	85.15	28.93	33.11	28.92	33.10
-BnTQA	–	–	–	–	–	–	–	–	92.40	88.10	92.42	88.12	**41.32**	**47.26**	**41.32**	**47.26**

Table 1: Baseline, BnTQA-X and HiTQA-X models' scores. -X represents the backbone architecture of a fine-tuned model and – entries are for incompatible models in a low-resourced language (Bengali or Hindi).

is the percentage of correctly predicted rows among all target rows.

(3) **Column Exact Match F1 (Col)**: Column EM precision is the percentage of correctly predicted columns and corresponding headers among all predicted columns. Column EM recall is the percentage of correctly predicted columns among all target columns.

(4) **Cell Exact Match F1 (Cell)** is the most relaxed metric. Cell EM precision is the percentage of correctly generated cells among all predicted cells. Cell EM recall is the percentage of correctly predicted cells among all target cells.

6 Results

Baselines. As reported in Table 1, `GPT-4` performs the best on our test set with a table EM accuracy of 26.83%. `GPT-3.5` under-performs `GPT-4` but is better than open-sourced LLMs. Open-source LLMs, `OdiaG` is pre-trained on Bengali text data but not on structured table data. The low accuracy of `OdiaG` (0.69%) can be attributed to the models' lack of table understanding and table specific question which differs significantly from text-based tasks on which it has been pre-trained on as shown in examples in Appendix A.6. Baseline encoder-decoder model, `En2Bn`, fine-tuned on translated tableQA data, correctly generates 4.73% of rows and cells and under-performs `OdiaG`, but is better than `TableLlama`. Although fine-tuning improves Bengali understanding, the low scores can be attributed to the erroneous translations of English tables in the MultiTabQA dataset which corroborate with (Minhas et al., 2022) that table translation leads to error-propagation to down-stream QA task. Further, a lack of culture-specific tables in the MultiTabQA pre-training dataset leads to downgraded performance on topics in the BanglaTabQA test set. In conclusion, `GPT-4` is able to perform table reasoning in low-resourced Bengali, but is very expensive and closed-source, limiting it's accessibility and utility. `GPT-3.5`'s and all open-access baseline models' low scores demonstrates the need for both task and language adaptation with a large-scale dataset for training accessible open-source language models for low-resourced tableQA.

BanglaTabQA models. Parameter-efficient fine-tuned Llama models, `BnTQA-llama`, achieves comparable results to `GPT-3.5`. Table 1 shows that fine-tuned encode-decoder models, `BnTQA-mBart` and `BnTQA-M2M`, outperforms `GPT-4` on table exact match accuracy (EM) and column EM F1, but not for row and cell EM F1. This can be attributed to incorrect header generation of `GPT-4` reflecting in column and subsequently table EM scores. Apart from `GPT-4`, all other baseline models underperform BanglaTabQA encoder-decoder models by a large margin on all metrics. `BnTQA-llama` overfits to the validation set, and does not generalize well to the test set. The low scores of PEFT compared to full fine-tuning (FT) can be attributed to insufficient alignment of the frozen parameters of the backbone Llama model and sub-optimal tokenization of Bengali which has been observed in SentencePiece tokenizers in non-Latin languages (Banerjee and Bhattacharyya, 2018; Cui et al., 2023). The results establishes the quality of the BanglaTabQA dataset and its effectiveness in adapting neural models to *both* language and table understanding.

HindiTabQA models. We follow a similar experimental setup as discussed in Section 5. We report the results in Table 1. We observe that `HiTQA-BnTQA`, initialized with with `BnTQA-mbart`, outperforms all HindiTabQA models and achieves

Model	No post-processing				With post-processing			
BnTQA	**Tab**	**Row**	**Col**	**Cell**	**Tab**	**Row**	**Col**	**Cell**
-llama	0.00	0.00	0.00	0.26	5.74	17.59	5.69	15.49
-mBart	0.00	8.70	10.74	8.70	19.01	20.74	19.01	20.74
-M2M	0.00	0.00	0.00	0.00	18.18	35.80	18.18	35.80

Table 2: Zero-shot cross-lingual transfer scores of BnTQA models on Hindi test data.

a test score of 41.32%. Similar to BanglaTabQA, `HiTQA-mBart` outperforms `HiTQA-M2M` with a table EM test score of 33.06% and 28.93% respectively. `HiTQA-llama` underperforms compared to the encoder-decoder models. All models trained on the HindiTabQA dataset outperform the two-shot in-context learning baseline models. The results follow a similar trend to BanglaTabQA models and prove that our data generation process is generalizable and the HindiTabQA dataset is able to align neural models for tableQA task in Hindi.

6.1 Zero-shot Cross-lingual Transfer

We further study generalizability, by selecting the best performing language, Bengali, and evaluating the BanglaTabQA models on Hindi test set in a zero-shot setting *without* training on Hindi data. This setup allows us to study the cross-lingual transfer of BanglaTabQA models to Hindi with a different script, and evaluate how well the models generalize to new out-of-distribution input tables. BanglaTabQA models are able to perform table reasoning in Hindi indicating semantic information transfer across languages. We demonstrate some examples in the Appendix A.7. Table headers and numbers generated from math operations are often in Bengali instead of Hindi (Example 7). Extractive questions are generated correctly (Example 8). Table 2 lists the zero-shot cross-lingual scores using the original predictions (named "No Post-Processing") of the BanglaTabQA models on the Hindi test set defined in Section 4.7. Additionally, we perform post-processing of the predictions to translate the predicted tables' values to Hindi. As translating tables, composed of numbers and entities, with machine translation systems is unreliable (Minhas et al., 2022), we follow an automatic post-processing pipeline to transform predicted answer tables to Hindi. First, all lexical occurrence of Bengali digits in predictions are replaced with Hindi digits using a dictionary. Next, all lexical occurrence of SQL keyword in Bengali in the prediction headers are replaced with a Bengali-to-SQL keyword mapping and subsequently with a SQL-to-Hindi mapping described in Section 4. This fixes most of the Bengali presence in the predictions. Finally, we translate the predicted column names/values in Bengali to Hindi with Google translate. Table 2 shows that post-processing increases the scores, demonstrating the generalizability of BanglaTabQA models' table reasoning capabilities on out-of-domain Hindi tables with unseen cultural entities. This further demonstrates the quality and utility of the BanglaTabQA dataset and our proposed data generation method and quality of the trained models.

6.2 Mathematical Operator classes

We study how BanglaTabQA and HindiTabQA datasets aid in Bengali and Hindi numeracy and math understanding by evaluating `BnTQA-mBart` and `HiTQA-mBart` on 6 categories of operator classes (Section 4.5). We observe in Table 4 that `BnTQA-mbart` performs best on *groupBy (G)* operators with a table EM accuracy of 50.00% and `HiTQA-mBart` on *Sorting (So)* operators with a table EM accuracy of 39.05%. Both models are able to generalize to unseen tables in the respective languages' test sets. This affirms that BanglaTabQA and HindiTabQA dataset aids mathematics reasoning of the trained models and enhances numeracy understanding in the low-resourced language.

7 Conclusion

Our work introduces tableQA for the low-resource languages. We propose a methodology for large-scale dataset development on limited budget and automatic quality control which can be applied over any low-resource language with a web-presence. We discuss in detail the application of the methodology with an Indic Language, Bengali, for which we release a large-scale dataset, BanglaTabQA. We further demonstrate generalizability of the process with another language, Hindi. We assess the datasets' quality by effectively training different Bengali and Hindi tableQA models and conducting various experiments on model efficacy. Our studies on different operator classes and zero-shot cross-lingual transfer demonstrate that models trained with our dataset generalize well to unseen tables. Our proposed methodology can promote further research in low-resource tableQA, while our released dataset and models can be used to further explore tableQA for Bengali and Hindi.

Operator class	Operations
arithmetic (A)	count, sum, average, max, min
sorting (So)	ascending, descending
groupBy (G)	table column/row grouping
filtering (F)	where, having
set (Se)	union, intersect, except
logical (L)	and, or, not, in, not in, between

Table 3: Classification of tableQA operations.

Op	Bengali				Hindi			
	Tab	Row	Col	Cell	Tab	Row	Col	Cell
A	39.66	55.64	39.67	55.64	35.06	41.71	35.07	41.71
So	25.00	25.00	25.00	25.00	**39.05**	**42.74**	**39.05**	**42.74**
G	**50.00**	**76.92**	**50.00**	**76.92**	33.33	35.96	33.33	35.96
F	37.78	35.86	37.77	35.86	23.23	26.35	23.23	21.67
Se	36.11	49.10	36.11	49.10	5.00	11.11	5.00	11.11
L	34.38	13.23	34.38	13.23	25.58	27.38	25.58	27.38

Table 4: XTQA-mBart test set scores (%) on Operator Class (Op); X is a low-resourced language (Bn or Hi).

Limitations

We design a scalable automatic tableQA data generation method and apply it on with two low-resourced languages: Bengali and Hindi. We release two tableQA datasets: BanglaTabQA and HindiTabQA and several models as outcome. Our main results in Table 1 demonstrate successful adaptation of neural models to low-resourced tableQA task. Our extensive experimentation on generalizability in Section 6.1 and 6.2 shows that models trained on the BanglaTabQA dataset performs well across all operator classes and generalize to unseen languages and tables, proving generalizability of the datasets and methodology.

Our dataset methodology is generalizable, but it is limited to languages for which unlabelled tables are available online. For very-low resource languages with low web presence, our method has only limited impact. Also, we used SQUALL templates for query generation, which do not support multi-table operations or complex queries. We leave addressing these challenges to future work.

Ethical Considerations

The task and models proposed in the paper is aimed at closing the gap of resource scarcity in low-resource languages. To do so, we have used existing open-source resources publicly available in the web under MIT, CC-BY-SA-3.0 and MIT, CC-BY-SA-4.0 licenses. Our dataset is generated synthetically data and will be released under MIT, CC-BY-SA-4.0 license. Our synthetic samples use templates from the SQUALL dataset also released under MIT, CC-BY-SA-4.0 license. Our test data splits are manually annotated. We pay each annotator €13.27/hour for their efforts. Further, we have utilized Wikipedia tables from Huggingface Wikipedia dataset. Wikipedia tables contain information about named-entities, facts and events in the public domain. We do not use any user-specific or sensitive data and information. Our models are built over open-source encoder-decoder models and closed-source GPT-3.5. Our work did not explicitly handle any bias which exists in the aforementioned pre-trained models or Wikipedia.

Acknowledgements

We thank Elsevier's Discovery Lab for their support throughout this project and funding this work. This work was also supported by Dutch Research Council (NWO), under project numbers 024.004.022, NWA.1389.20.183, KICH3.LTP.20.006, and VI.Vidi.223.166, and the European Union's Horizon Europe program under grant agreement No 101070212. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

References

Tamali Banerjee and Pushpak Bhattacharyya. 2018. Meaningless yet meaningful: Morphology grounded subword-level NMT. In *Proceedings of the Second Workshop on Subword/Character LEvel Models*, pages 55–60, New Orleans. Association for Computational Linguistics.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack

Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Zhoujun Cheng, Haoyu Dong, Ran Jia, Pengfei Wu, Shi Han, Fan Cheng, and Dongmei Zhang. 2022. FORTAP: Using formulas for numerical-reasoning-aware table pretraining. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1150–1166, Dublin, Ireland. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. PaLM: Scaling language modeling with pathways.

Yiming Cui, Ziqing Yang, and Xin Yao. 2023. Efficient and effective text encoding for Chinese LLaMA and Alpaca. *arXiv preprint arXiv:2304.08177*.

Raj Dabre, Himani Shrotriya, Anoop Kunchukuttan, Ratish Puduppully, Mitesh Khapra, and Pratyush Kumar. 2022. IndicBART: A pre-trained model for indic natural language generation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1849–1863, Dublin, Ireland. Association for Computational Linguistics.

Deborah A. Dahl, Madeleine Bates, Michael Brown, William Fisher, Kate Hunicke-Smith, David Pallett, Christine Pao, Alexander Rudnicky, and Elizabeth Shriberg. 1994. Expanding the scope of the ATIS task: The ATIS-3 corpus. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Arijit Das and Diganta Saha. 2022. Deep learning based bengali question answering system using semantic textual similarity. *Multimedia Tools Appl.*, 81(1):589–613.

Yashar Deldjoo, Johanne R. Trippas, and Hamed Zamani. 2021. Towards multi-modal conversational information seeking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 1577–1587, New York, NY, USA. Association for Computing Machinery.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2021. Beyond English-centric multilingual machine translation. *J. Mach. Learn. Res.*, 22(1).

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic bert sentence embedding. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76:378–382.

Jay Gala, Pranjal A Chitale, A K Raghavan, Varun Gumma, Sumanth Doddapaneni, Aswanth Kumar M, Janki Atul Nawale, Anupama Sujatha, Ratish Puduppully, Vivek Raghavan, Pratyush Kumar, Mitesh M Khapra, Raj Dabre, and Anoop Kunchukuttan. 2023. IndicTrans2: Towards high-quality and accessible machine translation models for all 22 scheduled Indian languages. *Transactions on Machine Learning Research*.

Vivek Gupta, Pranshu Kandoi, Mahek Vora, Shuo Zhang, Yujie He, Ridho Reinanda, and Vivek Srikumar. 2023. TempTabQA: Temporal question answering for semi-structured tables. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2431–2453, Singapore. Association for Computational Linguistics.

Tahmid Hasan, Abhik Bhattacharjee, Kazi Samin, Masum Hasan, Madhusudan Basak, M. Sohel Rahman, and Rifat Shahriyar. 2020. Not low-resource anymore: Aligner ensembling, batch filtering, and new datasets for Bengali-English machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2612–2623, Online. Association for Computational Linguistics.

Michael A. Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. 2021. A survey on recent approaches for natural language processing in low-resource scenarios. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2545–2568, Online. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333, Online. Association for Computational Linguistics.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training compute-optimal large language models.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

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

Sujay Kumar Jauhar, Peter D. Turney, and Eduard H. Hovy. 2016. Tables as semi-structured knowledge for question answering. In *Annual Meeting of the Association for Computational Linguistics*.

Zhengbao Jiang, Yi Mao, Pengcheng He, Graham Neubig, and Weizhu Chen. 2022. OmniTab: Pretraining with natural and synthetic data for few-shot table-based question answering. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 932–942, Seattle, United States. Association for Computational Linguistics.

Nengzheng Jin, Joanna Siebert, Dongfang Li, and Qingcai Chen. 2022. A survey on table question answering: Recent advances.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Md. Rezaul Karim, Sumon Kanti Dey, Tanhim Islam, Sagor Sarker, Mehadi Hasan Menon, Kabir Hossain, Md. Azam Hossain, and Stefan Decker. 2021. DeepHateExplainer: Explainable hate speech detection in under-resourced Bengali language.

Yannis Katsis, Saneem Chemmengath, Vishwajeet Kumar, Samarth Bharadwaj, Mustafa Canim, Michael Glass, Alfio Gliozzo, Feifei Pan, Jaydeep Sen, Karthik Sankaranarayanan, and Soumen Chakrabarti. 2022. AIT-QA: Question answering dataset over complex tables in the airline industry. *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*.

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

Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian guang Lou. 2021. TAPEX: Table pre-training via learning a neural SQL executor.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Bhavnick Minhas, Anant Shankhdhar, Vivek Gupta, Divyanshu Aggarwal, and Shuo Zhang. 2022. XInfoTabS: Evaluating multilingual tabular natural language inference. In *Proceedings of the Fifth Fact Extraction and VERification Workshop (FEVER)*, pages 59–77, Dublin, Ireland. Association for Computational Linguistics.

Linyong Nan, Chiachun Hsieh, Ziming Mao, Xi Victoria Lin, Neha Verma, Rui Zhang, Wojciech Kryściński, Hailey Schoelkopf, Riley Kong, Xiangru Tang, Mutethia Mutuma, Ben Rosand, Isabel Trindade, Renusree Bandaru, Jacob Cunningham, Caiming Xiong, and Dragomir Radev. 2022. FeTaQA: Free-form Table Question Answering. *Transactions of the Association for Computational Linguistics*, 10:35–49.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2023. Gpt-4 technical report.

Vaishali Pal, Evangelos Kanoulas, and Maarten de Rijke. 2022. Parameter-efficient abstractive question answering over tables or text. In *Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering*, pages 41–53, Dublin, Ireland. Association for Computational Linguistics.

Vaishali Pal, Andrew Yates, Evangelos Kanoulas, and Maarten de Rijke. 2023. MultiTabQA: Generating tabular answers for multi-table question answering. In *ACL 2023: The 61st Annual Meeting of the Association for Computational Linguistics*, pages 6322–6634.

Shantipriya Parida, Sambit Sekhar, Guneet Singh Kohli, Arghyadeep Sen, and Shashikanta Sahoo. 2023. Bengali instruction-tuning model. `https://huggingface.co/OdiaGenAI`.

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

Patti Price. 1990. Evaluation of spoken language systems: the ATIS domain. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Mahalakshmi J, Divyanshu Kakwani, Navneet Kumar, Aswin Pradeep, Srihari Nagaraj, Kumar Deepak, Vivek Raghavan, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Shantadevi Khapra. 2022. Samanantar: The largest publicly available parallel corpora collection for 11 Indic languages. *Transactions of the Association for Computational Linguistics*, 10:145–162.

Sebastian Ruder. 2019. The 4 biggest open problems in NLP. `https://www.ruder.io/4-biggest-open-problems-in-nlp`.

Tianze Shi, Chen Zhao, Jordan Boyd-Graber, Hal Daumé III, and Lillian Lee. 2020. On the potential of lexico-logical alignments for semantic parsing to SQL queries. In *Findings of EMNLP*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Yunhu Ye, Binyuan Hui, Min Yang, Binhua Li, Fei Huang, and Yongbin Li. 2023. Large language models are versatile decomposers: Decomposing evidence and questions for table-based reasoning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 174–184, New York, NY, USA. Association for Computing Machinery.

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. TaBERT: Pretraining for joint understanding of textual and tabular data. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921, Brussels, Belgium. Association for Computational Linguistics.

John M. Zelle and Raymond J. Mooney. 1996. Learning to parse database queries using inductive logic programming. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2*, AAAI'96, page 1050–1055. AAAI Press.

Liangyu Zha, Junlin Zhou, Liyao Li, Rui Wang, Qingyi Huang, Saisai Yang, Jing Yuan, Changbao Su, Xiang Li, Aofeng Su, Tao Zhang, Chen Zhou, Kaizhe Shou, Miao Wang, Wufang Zhu, Guoshan Lu, Chao Ye, Yali Ye, Wentao Ye, Yiming Zhang, Xinglong Deng, Jie Xu, Haobo Wang, Gang Chen, and Junbo Zhao. 2023. TableGPT: Towards unifying tables, nature language and commands into one GPT.

Weijia Zhang, Vaishali Pal, Jia-Hong Huang, Evangelos Kanoulas, and Maarten de Rijke. 2024. Qfmts: Generating query-focused summaries over multi-table inputs.

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

Yilun Zhao, Zhenting Qi, Linyong Nan, Boyu Mi, Yixin Liu, Weijin Zou, Simeng Han, Ruizhe Chen, Xiangru Tang, Yumo Xu, Dragomir Radev, and Arman Cohan. 2023a. QTSumm: Query-focused summarization over tabular data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1157–1172, Singapore. Association for Computational Linguistics.

Yilun Zhao, Zhenting Qi, Linyong Nan, Boyu Mi, Yixin Liu, Weijin Zou, Simeng Han, Xiangru Tang, Yumo Xu, Arman Cohan, and Dragomir Radev. 2023b. QTSumm: A new benchmark for query-focused table summarization.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2SQL: Generating structured queries from natural language using reinforcement learning.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3277–3287, Online. Association for Computational Linguistics.

A Appendix

A.1 Bengali SQL2NQSim (LaBse fine-tuning) Results

We evaluate semantic similarity of the LaBse model trained on the translated semantic parsing datasets comprising of Bengali SQL and it corresponding Bengali question (Section 4.4) and report the validation set results in Table 5. Both datasets show high semantic similarity among query-question pairs. However, BanglaTabQA have a higher semantic similarity on various distance metrics indicating higher similarity of the query-question pairs compared to HindiTabQA. HindiTabQA lower semantic scores can be attributed to the lower recall scores among query-question pairs leading to lower F1 similarity scores.

Scores	Bengali	Hindi
Accuracy with Cosine-Similarity	91.99	98.67
F1 with Cosine-Similarity	92.30	72.16
Precision with Cosine-Similarity	94.55	77.68
Recall with Cosine-Similarity	90.15	67.36
Avg Precision with Cosine-Similarity	97.79	75.32
Accuracy with Manhattan-Distance	91.97	98.62
F1 with Manhattan-Distance	92.31	70.96
Precision with Manhattan-Distance	93.73	77.15
Recall with Manhattan-Distance	90.94	65.69
Avg Precision with Manhattan-Distance	97.80	74.41
Accuracy with Euclidean-Distance	91.99	98.67
F1 with Euclidean-Distance	92.30	72.16
Precision with Euclidean-Distance	94.55	77.68
Recall with Euclidean-Distance	90.15	67.36
Avg Precision with Euclidean-Distance	97.79	75.32
Accuracy with Dot-Product	91.99	98.67
F1 with Dot-Product	92.30	72.16
Precision with Dot-Product	94.55	77.68
Recall with Dot-Product	90.15	67.36
Avg Precision with Dot-Product	97.79	75.32

Table 5: Bengali SQL2NQSim validation scores (%)

A.2 Bengali SQL2NQ model Results

We report the validation scores of the SQL2NQ models in Table 6. The Bengali SQL2NQ model scores are lower than the Hindi SQL2NQ model. Manual inspection of the generated dataset reveals that the Hindi questions and query have higher lexical overlap compared to the Bengali questions-query pairs where the questions are more natural leading to lower lexical overlap with the corresponding SQL query.

A.3 Open-Source Backbone Model Size

We used the following open-source models as backbone to low-resource tableQA task. As observed in Table 7, `M2M_418` is the smallest backbone model among all models and `Llama-7b` is the largest.

	Bengali	Hindi
Rouge-1	14.63	53.20
Rouge-2	5.83	24.98
Rouge-L	14.28	51.58

Table 6: Bengali SQL2NQ model's validation scores (%)

Model	Number of Parameters
mbart-large-50	0.680 billion
m2m100_418M	0.418 billion
Llama-7B	7 billion

Table 7: Backbone model sizes

A.4 GPT Prompts

The 2-shot in-context learning prompt with demostrations to GPT is shown in Prompt A.1:

Prompt A.1: 2-Shot ICL Prompt for GPT-3.5/4

আপনি একজন সহায়ক সহকারী যিনি বাংলা প্রশ্নের উত্তর দেন বাংলা টেবিল থেকে বাংলায় উত্তর টেবিল তৈরি করে। m সারি এবং n কলামগুলির একটি টেবিল নিম্নলিখিত প্যাটার্নে লেখা হয়ে: <কলাম> টেবিল হেডার <রো ১> মান ১,১ । মান ১,২ । ... মান ১,n <রো ২> মান ২,১ । ... <রো m> মান m,১ । মান m,২ । ... । মান m,n

উদাহরণ:

১) প্রশ্ন: কটা শিরোনাম কাউন্টডাউন? <কলাম> বছর । শিরে-নাম । ভুমিকা <রো ১> 2006 । সি নো ইভল । জেকব গুড নাইট ...<রো ১৩> 2016 । কাউন্টডাউন । লেঃ ত্রোনিন <রো ১৪> 2016 । কাউন্টডাউন । লেঃ ত্রোনিন <রো ১৫> 2016 । কাউন্টডাউন । লেঃ ত্রোনিন

উত্তর: <কলাম> গণনা('শিরেনিাম') <রো ১> ৩

২) প্রশ্ন: কটা বছরে শিরেনিাম সী নো ইভল? <কলাম> বছর । শিরোনাম । ভুমিকা <রো ১> 2006 । সি নো ইভল । জেকব গুড নাইট <রো ২> 2006 । সি নো ইভল । জেকব গুড নাইট <রো ৩> 2006 । সি নো ইভল । জেকব গুড নাইট ...

উত্তর: <কলাম> গণনা('বছর') <রো ১> ৩

The English translation of the 2-shot prompt for in-context learning (ICL) of GPT-3.5/4 is shown in

Prompt A.2:

Prompt A.2: 2-Shot ICL Prompt for GPT-3.5/4 (English translation)

You are a helpful assistant who answers Bengali questions from Bengali tables by generating an answer table. A table of m rows and n columns is written in the following pattern: <column> table header <row 1> value 1,1 | value 1,2 | ... value 1,n <row 2> value 2,1 | ... <row m> value m,1 | value m,2 | ... | value m,n
Examples:
1) **Question:** How many titles are Countdown? <column> year | Title | Role <row 1> 2006 | See No Evil | Jacob Go ... <row 13> 2016 | Countdown | Le Trunin <row 14> 2016 | Countdown | Le Trunin <row 15> 2016 | Countdown | Le Trunin
Answer: <column> count('Title') <row 1> 3
2) **Question:** How many years have See no Evil as titles? <column> year | Title | Role <row 1> 2006 | See No Evil | Jacob Good Night <row 2> 2006 | See No Evil | Jacob Good Night | <row 3> 2006 | See No Evil | Jacob Good Night ...
Answer: <column> count('year') <row 1> 3

A.5 LLama-based model Model Prompt

The 2-shot in-context learning prompt with demostrations to Llama-7B based model, OdiaG, is shown in Prompt A.3:

Prompt A.3: 2-Shot ICL Prompt for odiagenAI-bn

Instruction:

আপনি একজন সহায়ক সহকারী যিনি বাংলা টেবিল তৈরি করে বাংলা প্রশ্নের উত্তর দেন। উদাহরণ:

###Input:

কটা শিরোনাম কাউন্টডাউন? <কলাম> বছর | শিরোনাম | ভূমিকা <রো ১> 2014 | সী নো এভল ২ | জেকব গুড নাইট <রো ২> 2016 | কাউন্টডাউন | লেঃ ত্রেুনিন <রো ৩> 2016 | কাউন্টডাউন | লেঃ ত্রেুনিন

Response:

<কলাম> গণনা(শিরোনাম) <রো ১> ২

###End

###Input:

কটা বছর শিরোনাম সী নো এভল ২? <কলাম> বছর | শিরোনাম | ভূমিকা <রো ১> 2014 | সী নো এভল ২ | জেকব গুড নাইট <রো ২> 2016 | কাউন্টডাউন | লেঃ ত্রেুনিন <রো ৩> 2016 | কাউন্টডাউন | লেঃ ত্রেুনিন

Response:

<কলাম> গণনা(শিরোনাম) <রো ১> ১

###End

###Input:
{input}

Response:

The English translation of the 2-shot in-context learning prompt with demostrations to Llama-7B based model, OdiaG, is shown in Prompt A.4:

Prompt A.4: 2-Shot ICL Prompt for odiagenAI-bn (English translation)

Instruction:

You are a helpful assistant who generates answers Bengali table to answer Bengali questions. Examples:

###Input:

How many titles are Countdown? <column> year | Title | Role <row 1> 2014 | See No Evil 2 | Jacob Goodnight <row 2> 2016 | Countdown | Le Trunin <row 3> 2016 | Countdown | Le Trunin

###Response:

<column> count(Title) <row 1> 2

End

###Input:

How many years have See no Evil as titles? <column> year | Title | Role <row 1> 2014 | See No Evil 2 | Jacob Goodnight <row 2> 2016 | Countdown | Le Trunin <row 3> 2016 | Countdown | Le Trunin

Response:

<column> count(year) <row 1> 1

###Input:
{input}

###Response:

A.6 BnTabQA Models Qualitative analysis

We analyze the output of each model with an example to identify error patterns and factors that impact model predictions. The test set question কার নামে ফুটসাল সমন্বয়কারী অথবা প্রযুক্তিগত পরিচালকের অবস্থান আছে? (Who has the position of Futsal Coordinator or Technical Director?), involves logical operator `or` after extracting values for ফুটসাল সমন্বয়কারী (Fulsal Coordinator) and প্রযুক্তিগত পরিচালকের (Technical Director) from the column অবস্থান (`Position`). The input table is shown in Table 8 (translation of each table cell is italicized and in parenthesis for non-native readers) with target (English translation italicized and in parenthesis):

নাম (*Name*))
মাইকেল সকুবালা (*Michael Skubala)*
লেস রিড (*Les Reed)*

Example 1. Baseline encoder-decoder model, `En2Bn`, fine-tuned on the translated MultiTabQA dataset, correctly extracts মাইকেল স্কুবালা (*Michael*

অবস্থান *(Position)*	নাম *(Name)*
সভাপতি *(Chairman)*	গ্রেগ ক্লার্ক *(Greg Clark)*
সহ-সভাপতি *(Co-Chairman)*	ডেভিড গিল *(David Gil)*
সাধারণ সম্পাদক *(General Secretary)*	মার্ক বুলিংহ্যাম *(Mark Bullingham)*
কোষাধ্য *(Treasurer)*	মার্ক বারোস *(Mark Burroughs)*
গণমাধ্যম এবং যোগযোগ পরিচালক *(Media and Communications Director)*	লুইসা ফিয়ান্স *(Louisa Fiennes)*
প্রযুক্তিগত পরিচালক *(Technical Director)*	লেস রিড *(Les Reed)*
ফুটসাল সমন্বয়কারী *(Futsal Coordinator)*	মাইকেল স্কুবালা *(Michael Skubala)*
জাতীয় দলের কোচ (পুরুষ) *(National Team Coach (Male))*	গ্যারেথ সাউথগেট *(Gareth Southgate)*
জাতীয় দলের কোচ (নারী) *(National Team Coach (Female))*	ফিল নেভিল *(Phil Neville)*
রেফারি সমন্বয়কারী *(Referee Coordinator)*	নিল ব্যারি *(Neil Barry)*

Table 8: Example: BnTabQA Input Table. (English translation of each cell is italicized and in parenthesis)

Skubala) as the ফুটসাল সমন্বয়কারী (Fulsal Coordinator), but wrongly assigns it as the table header instead of নাম (name). Moreover, it generates the same entity twice instead of generating লেস রিড (Les Reed):

ফুটসাল সমন্বয়কারী *(Futsal Coordinator)*
মাইকেল সকুবালা *(Michael Skubala)*
মাইকেল সকুবালা *(Michael Skubala)*

Example 2. `OdiaG` also overfits to the demonstrations with গণনা (count) operator to generate incorrect value and header:

গণনা('নাম') *(count(Name))*
১ *(1)*

Example 3. `GPT-3.5` with 2-shot in-context learning (ICL) extracts মাইকেল স্কুবালা *(Michael Skubala)* correctly but generates an incorrect table header over-fitting to the demonstrations:

গণনা('নাম') *(count(Name))*
মাইকেল সকুবালা *(Michael Skubala)*

Example 4. `GPT-4` with 2-shot in-context learning (ICL) correctly generates the answer table:

নাম *(Name)*
মাইকেল সকুবালা *(Michael Skubala)*
লেস রিড *(Les Reed)*

Example 5. Both encoder-decoder models, `BnTQA-mBart` and `BnTQA-M2M`, fine-tuned on BanglaTabQA dataset, correctly generates both answer table headers and values:

নাম *(Name)*
মাইকেল স্কুবালা *(Michael Skubala)*
লেস রিড *(Les Reed)*

Example 6. `BnTQA-Llama`, fine-tuned on BanglaTabQA dataset, is partially correct in its predictions by generating ফুটসাল সমন্বয়কারী (Fulsal Coordinator) in the first row, but incorrectly repeats the same entity instead of লেস রিড (*Les Reed)* in the second row:

নাম *(Name)*
ফুটসাল সমন্বয়কারী *(Fulsal Coordinator)*
ফুটসাল সমন্বয়কারী *(Fulsal Coordinator)*

We observe from the examples that all baselines except `GPT-4` generate wrong table headers and overfits and mimics the demonstrations, showing a lack of understanding of table structure and reasoning. The BanglaTabQA models perform table reasoning, reflecting the utility and quality of the large-scale BanglaTabQA dataset.

A.7 Zero-Shot Cross-Lingual Transfer Examples

Example 7. The Hindi question, वर्ष 2011 में कितने शीर्षक हैं? (*How many titles are there in year 2011?*), with Hindi input table, Table 9 (English translation is italicized and in parenthesis) and target table:

गणना (शीर्षक) *(count(Title))*
४ *(4)*

`BnTQA-mBart` correctly performs table reasoning but generates the answer in Bengali script instead of Devnagari (Hindi) script:

গনণা(শিরোনাম) *(count(Title))*
৪ *(4)*

Example 8. However, for Hindi extractive questions like कौनसे प्राप्तकर्ता अधिकतम बार आये हैं? (Which recipient occurs the maximum number of times?), with Hindi input table:

साल *(year)*	प्राप्तकर्ता *(Recipient)*
2016	विनोद भट्ट *(Vinod Bhatt)*
2016	विनोद भट्ट *(Vinod Bhatt)*
2017	तारक महेता [1] *(Tarak Mehta[1])*

and target table:

प्राप्तकर्ता *(Recipient)*
विनोद भट्ट *(Vinod Bhatt)*

`BnTQA-mBart` correctly generates the answer in Hindi:

प्राप्तकर्ता *(Recipient)*
विनोद भट्ट *(Vinod Bhatt)*

वर्ष (*year*)	शीर्षक (*Title*)	किरदार (*Character*)
2005	फ्लाइटप्लान (*Flight Plan*)	एरिक (*Eric*)
...	...	...
2011	इन टाइम (*In Time*)	हेनरी हैमिल्टन (*Henry Hamilton*)
2011	इन टाइम (*In Time*)	हेनरी हैमिल्टन (*Henry Hamilton*)
2011	इन टाइम (*In Time*)	हेनरी हैमिल्टन (*Henry Hamilton*)
2011	इन टाइम (*In Time*)	हेनरी हैमिल्टन (*Henry Hamilton*)
...	...	...
2014	स्पेस स्टेशन 76 (*Space Station 76*)	टैड (*Ted*)
...	...	...
2014	विंटर्स टेल (*Winter's Tale*)	पीटर लेक के पिता (*Peter Lake's Father*)

Table 9: Example: HiTabQA Input Table (English translation of each cell is italicized and in parenthesis)

A.8 Comparison of scores of LaBSE and SQL2NQ models

We qualitatively compare the sentence similarity models LaBse and SQL2NQ with examples shown in Table 10. We observe that LaBse scores are low for positive samples of Bengali SQL queries and the corresponding Bengali question. Further, negative samples, i.e., Bengali SQL query and an unrelated Bengali question has high similarity scores. This trend is not observed for the sentence similarity model, SQL2NQ, trained on Bengali SQL queries and corresponding Bengali natural questions.

	Bengali SQL	Bengali Question	LaBse Scores	SQL2NQ Scores
+ve	নির্বাচন করুন ‘বছর‘ দল করা ‘বছর‘ সাজান হোক গনণা(‘ফলাফল‘) সীমা ১ (SELECT years GROUP BY years ORDER BY COUNT(result) LIMIT 1)	কোন বছরে সবচেয়ে কম ফল হয়েছে? (Which year has the least number of results?)	0.45	0.94
	নির্বাচন করুন ‘শিরনাম‘ সাজান হোক ‘বছর‘ অবরোহী সীমা ১ (SELECT ‘title‘ ORDER BY ‘year‘ DESC LIMIT 1)	সম্প্রতিকতম বছরের সাথে সম্প্রতিক শিরনাম ফেরত দিন। (Return the most recent title of the most recent year?)	0.43	0.98
-ve	নির্বাচন করুন সর্বনিম্ন(‘সাল‘) (SELECT min(‘year‘))	কোন বছরে (২০১০, ২০১৬) সবচেয়ে বেতশ শংখ্যক পুর-ষ্কার জিতেছে? (In which year (2010, 2016) were the most number of awards received?)	0.51	0.31
	নির্বাচন করুন গনণা(*) যেখানে ‘কাজ‘=সৌদামিনীর সংসার (SELECT count(*) WHERE ‘work‘="The World of Saudamini")	মোট ৪ আছে এমন গেমের মোট শংখ্যা গনণা করুন। (How many games scored a total of 4?)	0.80	0.07

Table 10: Comparison of sentence similarity scores between LaBse and our trained SQL2NQ models.