TANQ: An Open Domain Dataset of Table Answered Questions

Mubashara Akhtar[*◇†‡] and **Chenxi Pang**[*†]
Andreea Marzoca[†] and **Yasemin Altun**[†] and **Julian Martin Eisenschlos**[†]
[◇]King's College London, UK & ETH Zurich, Switzerland
mubashara.akhtar@ai.ethz.ch
[†]Google DeepMind, Switzerland
{chenxipang,andreeam,altun,eisenjulian}@google.com

Abstract

Language models, potentially augmented with tool usage such as retrieval, are becoming the go-to means of answering questions. Understanding and answering questions in real-world settings often requires retrieving information from different sources, processing and aggregating data to extract insights, and presenting complex findings in form of structured artifacts such as novel tables, charts, or infographics. In this paper, we introduce TANQ,[1] the first open-domain question answering dataset where the answers require building tables from information across multiple sources. We release the full source attribution for every cell in the resulting table and benchmark state-of-the-art language models in open, oracle, and closed book setups. Our best-performing baseline, Gemini Flash, reaches an overall F1 score of 60.7, lagging behind human performance by 12.3 points. We analyze baselines' performance across different dataset attributes such as different skills required for this task, including multi-hop reasoning, math operations, and unit conversions. We further discuss common failures in model-generated answers, suggesting that TANQ is a complex task with many challenges ahead.

1 Introduction

Understanding and solving problems in real-world scenarios often requires reasoning across multiple documents and data modalities. This includes (i) retrieving information from different sources, (ii) processing and aggregating data to extract insights, and (iii) presenting complex findings in a structured format, for example a table or infographics, to communicate them to readers.

Previous studies show that knowledge workers across domains, e.g., finance, science, and economics, spend around 20% of their time searching and gathering information from different files into one document to extract insights and consequently answer information-seeking questions (Chui et al., 2012). As a result, one of most challenging tasks for workers is to aggregate data and turn it into insights. Tables as a structured representation of data are ubiquitous in real-world sources and are commonly used to communicate complex information. Hence, they can be the perfect modality to answer complex questions.

Large language models (LLMs), often enhanced with external tools, have become a primary method for various application such as answering questions. State-of-the-art question-answering (QA) systems integrate LLMs in various ways, from decomposing complex queries to retrieving documents using external tools or generating context data from knowledge acquired during model training. However, their evaluation is mostly limited to simple datasets, e.g., TabFact (Chen et al., 2020a) or HotpotQA (Yang et al., 2018), whose questions can be answered by reasoning over a single table or text document and generating a short text sequence as answer. This limits the applicability of such systems to perform complex multi-step research explorations. Moreover, it differs from real-world needs where relevant information can be spread across documents and represented in different forms (e.g., text or tables). More often than not, generating a short text as answer is not sufficient for complex information-seeking questions.

In this paper, we investigate LLMs' capabilities in reasoning over multiple data sources and formats (i.e., text, tables, infoboxes) to answer entity-centric questions and generate table answers as structured artifacts. To address these challenges, we introduce **TANQ**, an open-domain,

[*]Equal contributions.
[‡]Work done during Google DeepMind internship.
[1]Dataset available at github.com/google-deepmind/tanq.

Transactions of the Association for Computational Linguistics, vol. 13, pp. 461–480, 2025. https://doi.org/10.1162/tacl_a_00749
Action Editor: Mark Steedman. Submission batch: 4/2024; Revision batch: 10/2024; Published 6/2025.

Dataset	Open Domain	Multi Doc	Answer Type	Document Type		
				Text	Table	Infobox
InfoTabs (Gupta et al., 2020)			short text			✓
FeTAQA (Nan et al., 2022)			free form text		✓	
FinQA (Chen et al., 2021b)			short text	✓	✓	
TATQA (Zhu et al., 2021)			short text	✓	✓	
MultiHiertt (Zhao et al., 2022)			numeric	✓	✓	
OTTQA (Chen et al., 2021a)	✓		short text		✓	
NQ-TABLES (Herzig et al., 2021)	✓		short text		✓	
HybridQA (Chen et al., 2020b)		✓	short text	✓	✓	
MultiTabQA (Pal et al., 2023)		✓	table		✓	
TANQ	✓	✓	table	✓	✓	✓

Table 1: Comparison of TANQ to related (table) question-answering datasets.

multi-hop QA dataset. TANQ requires retrieving and aggregating data from multiple documents to compile and communicate answers as tables. To solve TANQ, models require different skills in addition to data retrieval such as filtering, maths, and name normalization. We create TANQ applying a five-step, automated data collection process. We use QAMPARI (Amouyal et al., 2022) as seed dataset and Wikidata as well as the Wikipedia corpus as data sources. For automated evaluation of different data collection and processing substeps, we use PaLM-2 (Anil et al., 2023b).

We evaluate several state-of-the-art LLMs on TANQ, including close, oracle and open book evaluation settings. Finally, we study model-generated answer tables and discuss common failure cases and challenges related to TANQ. Our evaluation of models across skills can further inform future tools and evaluation setups for LLMs to improve models for complex, information-seeking questions.

Our **contributions** are as follows:

(a) We introduce TANQ, the first open-domain question-answering benchmark that requires building answers in form of tables from multiple information sources.

(b) We benchmark state-of-the-art language models in oracle, open, and closed book setups, reaching an overall F1 score of 60.7 with our best-performing (oracle) baseline.

(c) We evaluate model performance across different dataset characteristics, and discuss challenges and common failure types.

2 Related Work

Various benchmarks for QA have been released in recent years. Each one addresses different challenges related to the task. Table 1 provides an overview and comparison of benchmarks.

QA with Text and/or Table Input. A number of datasets use text and one or multiple tables for QA. While both text and tables have been considered as input modalities, the output of the datasets is mostly limited to short textual answers. HybridQA (Chen et al., 2020b), for example, is a multi-hop QA dataset that requires reasoning over one table and multiple Wikipedia passages related to entities occurring in the table. HybridQA answers are short texts with location names being the most common answer types, followed by numbers, dates, and person names. Moreover, many QA benchmarks for reasoning over text and tabular context concentrate on the finance domain. For example, the MultiHiertt (Zhao et al., 2022) benchmark is created from financial reports. Questions require reasoning over texts and multiple tables. The answers are short numerical values with a focus on numerical reasoning. Other financial QA benchmarks are FinQA (Chen et al., 2021b) and TATQA (Zhu et al., 2021) where

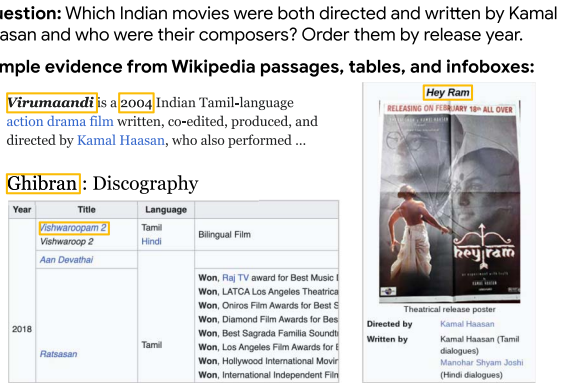
the context is one table and minimum two paragraphs related to the table. TATQA answers are short texts consisting of either one or multiple text spans from context paragraphs/tables or are free-form answers. MultimodalQA is a multi-hop, open-domain QA dataset that takes one table and related images and text paragraphs as input with answers similar to previously described datasets.

Open-domain Benchmarks. Most of the earlier described datasets have context provided in form of text and/or tables, whereas open-domain QA datasets first require extracting the relevant context, before answering the given question. The majority of open-domain QA datasets, such as WikiQA (Yang et al., 2015), TriviaQA (Joshi et al., 2017), and RobustQA (Han et al., 2023), are limited to textual context. NQ Tables (Herzig et al., 2021) extends table-QA to an open-domain setting where first top-k tables are retrieved from a given corpus. These tables are processed by a reader component for generating the correct short-text answer. Built on HybridQA, the Open Table-and-Text Question Answering (OTT-QA) benchmark (Chen et al., 2021a) extends this setting by requiring to extract both tables and texts given multi-hop questions.

QA with Table Answers. The work closest to ours with respect to input and output modalities is MultiTabQA (Pal et al., 2023). MultiTabQA seeds on the Spider dataset, a text-to-SQL dataset containing SQL queries, database tables, and natural language translations of the queries. Pal et al. (2023) use table names occurring in SQL queries to extract the input tables and query Spider databases for answer table generation. While the datas et also generates answers as tables, it has certain limitations: (i) the benchmark input is limited to tables; (ii) the input and output tables are highly structured database tables which differ from real-world scenarios where tables occur in documents and websites in various formats; (iii) the questions are limited to SQL-based queries.

3 Building the TANQ Benchmark

TANQ evaluates the capability to answer open domain, multi-hop questions by aggregating data and generating answer tables.



Movie	Composer	Release Year
Chachi 420	Vishal Bhardwaj	1998
Hey Ram	Ilaiyaraaja	2000
Virumaandi	Ilaiyaraaja	2004
Vishwaroopam	Shankar-Ehsaan-Loy	2013
Sabaash Naidu	Ilaiyaraaja	2016
Vishwaroopam II	Mohamaad Ghibran	2018

Figure 1: An example question in TANQ and its corresponding table answer. Supporting evidence from multiple pages in a Wikipedia snapshot is provided for each data point inside the table. We highlight the rationale inside each snippet in yellow. LLMs are evaluated with or without access to the evidence.

3.1 Task Definition

A TANQ dataset instance is a triple (q, t, D), consisting of an entity-centric question q, a table answer t, and a document set D (see Figure 1). To answer the multi-hop question q, first multiple *sub-answers* are extracted from the document set D. The answer is generated in table form $t = \{t_{i,j} | i \leq n, j \leq m\}$ consisting of n rows (i.e., one per extracted entities) and m columns. The documents in D provide supporting evidence for each cell of the answer table $t_{i,j}$. D is either provided as input to models (oracle setting) or retrieved from the Wikipedia corpus as D' (open book) and can consist of texts, tables, and infoboxes.

3.2 Preliminaries

QAMPARI. We use QAMPARI, which is an open-domain QA dataset with lists of entities as answers (Amouyal et al., 2022), as seed dataset. QAMPARI further includes Wikipedia text as supporting evidence for each entry of the answer list. Different from prior QA datasets with short textual answers, they align to natural questions which

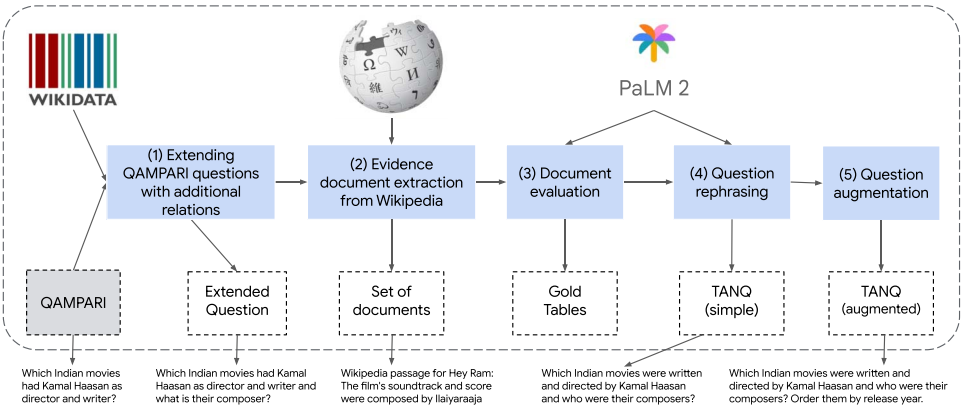


Figure 2: TANQ creation pipeline consisting of five steps: 1. Extending QAMPARI questions with additional relations based on Wikidata; 2. Evidence extraction (text, tables, infoboxes) from Wikipedia articles; 3. Evidence evaluation and (gold) answer table generation; 4. Rephrasing of question from first step; 5. Augmentation with additional skills to generate complex. We include a running example at the bottom.

require a list of answers extracted from multiple sources. The dataset is semi-automatically created with Wikidata and Wikipedia as data sources and evaluated by human annotators.

WikiData. Following Amouyal et al. (2022), we use Wikidata (WD) as a source for question generation. Wikidata (Erxleben et al., 2014) is a collaborative knowledge graph consisting of triples of entities and relations, i.e., (e_1, r, e_2). Entities e_i are values (e.g., 1990) or items that represent a real-world concept, object, or a topic.[2] Relations are edges connecting entities, e.g., `(DonaldTrump, instanceOf, human)`, where `Donald Trump` is the head entity and `human` the tail entity (Krishna et al., 2022). We follow the notion for formal queries over Wikidata introduced by Amouyal et al. (2022) for QAMPARI. Applying a relation r as query over WD items e_i results in a set of (tail) entities: $[[r(e)]] = \{e_i | (e_i, r, e) \in \text{WD}\}$.

Wikipedia. We extract supporting evidence in the form of sentences, tables and infoboxes (entity-tables at the top right corner of Wikipedia articles) from Wikipedia (WP).

3.3 TANQ Benchmark Pipeline

Figure 2 provides an overview of the TANQ pipeline outlining all steps for data collection, processing, and evaluation. We use QAMPARI, Wikidata, and Wikipedia, as well as PaLM-2 (Anil et al., 2023b) for paraphrasing and validation.[3]

[2] `https://en.wikipedia.org/wiki/Wikidata`.

[3] Prompts provided in the appendix (Figures 4 and 6).

Step 1. Extending QAMPARI Questions Starting with the QAMPARI questions and answer lists of entities, we extend each question q with additional WD relations using the answer entities e_a. QAMPARI questions are classified in either simple, composition (e.g., *''Who directed movies screen-written by Steven Spielberg?''*) or intersection (e.g., the example in Figure 1). We first query the WD knowledge graph to extract additional relations r_{ext} linked to e_a, i.e., $r[e_a] = \{e_{ext} | (e_a, r_{ext}, e_{ext}) \in \text{WD}\}$. Hence, each extension is a WD triple linking the QAMPARI answer e_a (e.g., *Hey Ram* in Figure 1) through the relation r_{ext} (i.e., *composer*) to a new extended entity e_{ext}, i.e., *Ilaiyaraaja* for relation *composer* and answer entity *Hey Ram* in Figure 1. We only select a relation which fulfils two conditions. First, it is part of a predefined relation set R, i.e., $r_{ext} \in R$. We manually specify R based on the WD relation used to create QAMPARI questions. Second, the relation exists for all answers e_a of question q. Given n extension relations, which fulfil these conditions, we extend the question q in a template-style fashion: ''$[q]$ and what is their $[r_{ext1}]$, $[r_{ext2}]$, [...] and $[r_{extn}]$.'' For example, the question in Figure 1, was generated based on the initial question *''Which Indian movies were both directed and written by Kamal Haasan?''* through extending with the WD relations *composer* and *release year*. Additionally, we extract all extension entities e_{ext} of the extension triple.

Step 2. Evidence Extraction from Wikipedia Next, we collect for each extension triple (e_a,

Question Type	Example Question
1. Simple	Which Belgian Grand Prix did Michael Schumacher win, and when did that happen?
2. Intersection	For which movie did Chris Columbus receive credits as **both director and writer** and what was their composer, publication year, duration, and genre?
3. Composition	**Who choregraphed a work that was produced by** the Royal Ballet? What was their date of birth, place of birth, occupation, and which awards did they receive?
Skill	
1. No skill	What filmmaker directed a movie written by Val Guest and what is their place of birth, date of birth, occupation, date of death, and place of death?
2. Filtering numeric	Which film was directed and produced by Mel Brooks and what was their composer and duration? **Filter the answer table for duration equal to or larger than 88 minutes.**
3. Filtering time	Which Italian footballer transferred to Pro Sesto in summer of 2020 and what is their date of birth, place of birth? **Filter the answer table for date of birth equal to or after 1985.**
4. Filtering entity	What pieces of writing did Gregory Benford edit? What were their publication dates and publishers? **Filter the answer table for publisher equal to Bantam Books.**
5. Date-to-year conversion	Who were the members of Black Sabbath and what was their **year of birth**, genre, instrument, and occupation?
6. Quantity conversion	What work did Michael Mann write and direct and what was their publication date, **duration in hours**, genre, director of photography?
7. Time calculation	Which governor of Connecticut died while in office? What was their place of birth, occupation, date of death, date of birth, political party and **how many years did they live**?
8. Approximation	What are the townships in Harper County, Kansas and what is their population **rounded to the nearest ten**?

Table 2: An overview of TANQ question types and skills we use for augmenting the questions for more complex reasoning in the final step of the TANQ pipeline in Figure 2.

r_{ext}, e_{ext}) supporting evidence using Wikipedia as an evidence source. Hereby, we search for supporting text, tables and infoboxes in the WP articles of e_a and e_{ext}. We apply simple heuristics and search for mentions of e_a in the e_{ext} article and vice versa. Moreover, we extend our queries with additional (heuristic-based) query words, for example, considering different formats of how numbers and dates are represented in queries.

Step 3. Evidence Evaluation & Answer Table Generation To evaluate the correctness of the previously collected evidence texts, tables, and infoboxes, we employ PaLM-2 as an evidence evaluator. We prompt the LLM to evaluate the extracted evidence in a natural language inference setting. For each extension triple (e_a, r_{ext}, e_{ext}), we construct template-based sentences s: ''$\langle e_a \rangle \langle r_{ext} \rangle \langle e_{ext} \rangle$'', e.g., *''Hey Ram composer Ilaiyaraaja''* for row 2 in Figure 1. We query the LLM to label the sentence s as ''supported'', ''refuted'', ''not enough information'' based on the provided evidence in form of a sentence, table or infobox entry. We then only consider the triples supported by at least one piece of evidence.[4]

Answer Table. Each extended relation corresponds to a column in the answer table (e.g., column *''Composer''* in Figure 1). Since the question in Figure 1 is extended with two additional relations (i.e., *composer* and *release year*), the resulting answer table has three columns. Hence, each cell in the answer table corresponds to a WD triple. For example, `(Hey Ram, composer, Ilaiyaraaja)` for cell *''Ilaiyaraaja''*. Some generated answer tables contained multiple entries in a single cell, as seen in the `genre` column of Table 7. To generate realistic tables, we filtered out samples with more than five entries in any single cell, resulting in a test set of 1,074 TANQ samples for evaluation.

Step 4. Question Rephrasing This step increases the naturalness of template-based extension questions generated in Step 1. Similarly to

[4]See Figure 4 for the exact prompt we used.

Type	#	%	Type	#	%
WD Item	57.6k	79.1	Numeric	3.2k	4.4
Time	11.9k	16.4	Text	46	0.1

Table 3: Types of Wikidata entities we use in the first pipeline step (see Figure 2) to extend QAMPARI and generate TANQ questions. We mostly use Wikidata (WD) items.

the earlier step, we prompt a PaLM-2 model in a few-shot setting. To ensure the question meaning is preserved during rephrasing, we add structured annotations in parenthesis to questions with the name of each relation (e.g., *''Which Indian movies were both directed (directed_by) and written by (written_by) Kamal Haasan?''*). We run up to 5 iterations of rephrasing and stop if all relations are present, or discard the question otherwise.[5]

Step 5. Augmenting with Skills Finally, to generate more challenging questions requiring further reasoning capabilities beyond retrieval, we extend TANQ questions by asking for realistic post-processing steps. Overall, we augment questions with the following additional skills: (i) **Filtering** of answer table given a numeric, time, or attribute condition (rows 2–4 in Table 2); (ii) **Conversion** of numbers, dates, locations and corresponding units (rows 5–7 in Table 2); (iii) **Calculation** and introduction of an additional table column based on time attributes (e.g., *''lifespan''* in row 8 in Table 2). (iv) **Approximating** numbers to the nearest ten, hundred, etc. (see last row of Table 2). We use up to three distinct skills for augmenting TANQ questions. Table 4 provides a breakdown of TANQ dataset samples across skills.

3.4 Dataset Statistics and Analysis

Overall, the generated TANQ dataset has 1395 entries, 36.1% of question type *simple*, 40.9% *intersection*, and 22.9% *composition* questions. We further plan to release a TANQ training set with approximately $42k$ samples. Approximately 72.4% dataset instances (i.e., 1,010) require at least one additional skill to answer the question. See Table 4 for a breakdown of skills and question types in TANQ. TANQ questions have an average a length of 21 tokens and require extracting information about three relations on average. On average, TANQ answer tables have 6.7 rows and 4 columns. See Table 3 for WD entity types used for extending QAMPARI questions.

[5]See Figure 6 for the exact prompt we used.

Question Type	Count	Freq (%)
1. Simple	428	39.9
2. Composition	164	15.3
3. Intersection	482	44.9
Skill		
1. No skill	160	14.9
2. Filtering numeric	66	6.1
3. Filtering time	190	17.7
4. Filtering entity	268	25.0
5. Date-to-year conversion	230	21.4
6. Quantity conversion	64	6.0
7. Time calculation	23	2.1
8. Approximation	73	6.8
Skills per question		
1. No skill	300	27.9
2. One skill	657	61.2
3. Two skills	94	8.8
4. Three skills	23	2.1

Table 4: Question types and skills in TANQ. *Simple* denotes questions which require neither composition nor intersection. See Table 2 for exemplary questions.

3.5 Manual Evaluation of the Pipeline

We manually evaluated 100 TANQ samples to assess if noise is introduced through the dataset's automated generation pipeline. The focus of the analysis was twofold. First, examining the initial questions sourced from the QAMPARI dataset and, second, evaluating the generation of TANQ questions over the different pipeline steps. We identified five issues which are discussed below.

Propagation of QAMPARI Issues. While occurring only for a small subset of questions, one observation was that some errors present in the initial QAMPARI questions were propagated through the pipeline without being corrected. In 2 out of the 100 samples, the original QAMPARI questions contained issues, one was carried into TANQ while the other was fixed. Examples include: *''Robert Benton is the screenwriter and director of which software?''* This question confuses a film with software as we see by considering the requested attributes, i.e., screenwriter and director.

While these logical errors were carried forward into TANQ grammatical errors, for example, were fixed during rephrasing with LLMs: *''Black*

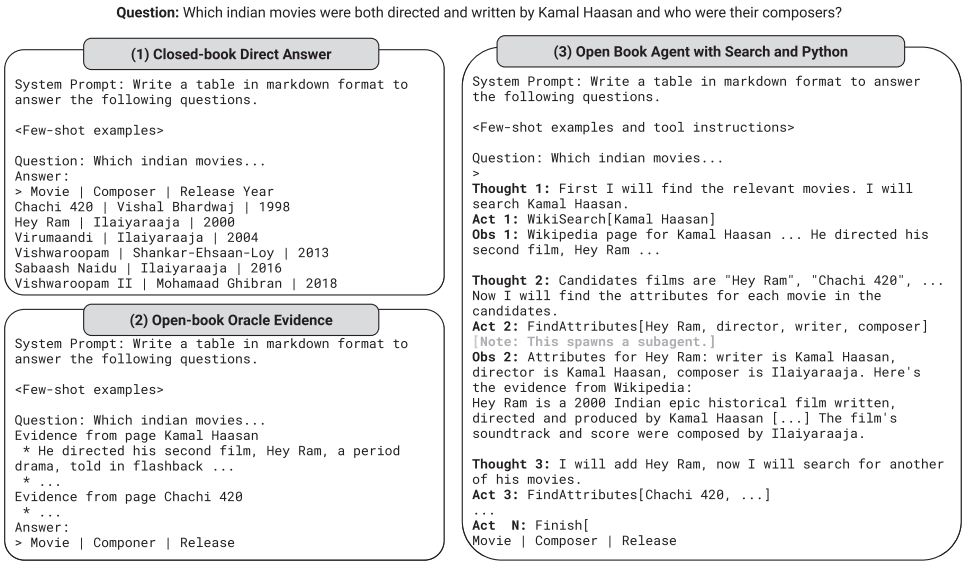


Figure 3: Prompts for baseline evaluation on TANQ: closed book, oracle, and open book with augmented tools.

Sabbath had who as a member?''—issues phrasing that was later improved to *''Who were the members of Black Sabbath [...]?''*.

Partially Revealed Answers. In 3 out of 100 evaluted questions, the question itself partially contained the answer, rendering the query less meaningful: *''Eon Productions and Harry Saltzman produced what series and what was their genre, director of photography, duration in second, producer, country of origin, composer, publication date, narrative location, screenwriter, and director?''*—in this case, the ''producer'' is already mentioned in the question.

Grammar and Expression Errors. Minor grammatical issues introduced through the TANQ pipeline were found in 2 out of the 100 samples. In one cases, this was fixed during the rephrasing process, while other remained unresolved. For example, *''+89 minute''* was rephrased to *''89 minutes''* in the sentence, *''Which film was directed and produced by Mel Brooks and what was their composer, genre, duration? Filter the answer table for duration equal to or larger than +89 minute.''* However, *''Larger than 89 minutes''* remained, but *''longer than 89 minutes''* would be more appropriate for this sentence.

Ordering of Relations. While our evaluation metrics is not sensitive to the ordering of columns within the answer table, the order of relation appeared confusing for some questions. This occurred for 7 out of the evaluated 100 examples as shown here: *''What filmmaker directed a movie written by Val Guest and what was their date of birth, place of death, date of death, and place of birth?''*—the order of personal attributes (i.e., date and place of birth, date and place of death) seems unnatural as date of birth would naturally be followed by place of birth.

Ambiguous or Unclear Relations. In 4 out of the 100 samples, certain relations were ambiguous or unclear, causing confusion about what exactly the question was asking for. Such as, *''Who directed the movie which had the screenwriting done by Marc Norman and what was their date of birth, language spoken, written or signed, place of birth, sex or gender, country of citizenship, occupation?''*—here, the attribute *''language spoken, written or signed''* is not clearly related to the director in question.

While we observed five different issues during the manual evaluation of the TANQ pipeline, these occurred only in a limited number of samples out of 100 evaluated ones. Thus, while these errors are present, they are not widespread enough to strongly impact the overall quality and utility of the TANQ dataset.

4 Baselines & Evaluation

We evaluate TANQ in (i) closed book, (ii) oracle, and (iii) open book setups. We give an overview of the different approaches in Figure 3.

Closed Book. The closed book setup evaluates LLMs' capabilities in extracting relevant information acquired during training to answer TANQ questions. For all experiments, we use the PaLM-2 Unicorn model (Anil et al., 2023b), GPT4*o* (Brown et al., 2020), Gemini Pro and Flash (Anil et al., 2023a), as well as Gemma 9B (Rivière et al., 2024). We evaluate all baselines in a few-shot setting with three examples provided in the prompt (see top left prompt in Figure 3).

Oracle. In the oracle setup, we provide models' source attribution for every cell of the answer table in form of oracle documents (i.e., text, tables, and infoboxes). Hence, the prompt consists of the TANQ question and multiple evidence sentences, infobox entries and/or tables (see bottom left prompt in Figure 3). To provide further context, we include the Wikipedia page title (for text evidence also sub-/section titles) where the evidence was extracted. Moreover, we add randomly selected oracle documents from other questions with similar attributes. This makes the oracle setting more challenging and requires models to filter the correct evidence from all provided ones first. We exclude models with a context length of less than $4k$ from the oracle evaluation. The evidence samples require a longer context to be fully displayed. Otherwise, performance may decrease due to the input being cut off.

Open Book. For the open book baseline, we extend the PaLM-2 Unicorn model with external tools for search and calculation. The Wikipedia-based search tools aim to mimic the human search approach on Wikipedia similar to Yao et al. (2023). The agent model can access Wikipedia information through three tools: (1) `WikiSearch(keywords)`: a keyword based Wikipedia search that returns the Wikipedia article most relevant to the provided keywords; (2) `FindEvidence(article, keywords)`: an article-specific search that returns matched sentences, tables and infobox entries given keywords and a Wikipedia article; (3) `GetIntro(article)`: returns the introduction section of a Wikipedia article. For calculations, we provide the model a Python engine as an external tool, (4) `Python(calculation)`. We design the agent model to decompose and solve the TANQ task in multiple sub-tasks, consisting of (i) retrieving requested entities in an iterative manner (e.g., movies in Figure 1), (ii) searching for entity-related information (e.g., release year), (iii) post-processing information (filtering, calculation, etc.), and finally *(iv)* aggregating the information in form of a table. For some sub-tasks (e.g., entity retrieval), a separate sub-agent augmented with the required tools (e.g., `WikiSearch`) is spawned.

Human Baseline. We compare model performance against human performance based on 100 answer tables generated by annotators. We provided the annotators the same input prompt as the models in the oracle setting (i.e., TANQ questions and evidence for each answer table cell).

Evaluation Metrics. To evaluate answer tables, we adopted a version of the *relative mapping similarity* (RMS) metrics introduced by Liu et al. (2023). RMS views tables as unordered collections of mappings from row/colum headers to values. Hence, the metric is invariant to transpositions and column/row permutations. It allows small errors between tables keys/values of target and reference tables using the Normalized Levenstein Distance (Biten et al., 2019). The metrics returns both precision and recall scores.

To evaluate the generated answer tables, we first converted table entries into a list of triplets. Each triplet consists of (i) the entity name (given in the first column of the answer table), (ii) the relation names (i.e., column names), and (iii) the content of a table cell. If a table cell contains multiple values, such as in column `Genre` in Table 7, the cell content is split into multiple triplets with the same entity name and attribute name, but with different values extracted from the cell. For example, given Table 7, the resulting splitted triplets contains multiple triplets for genre: {`(Evita, duration, 129 min), (Evita, genre, biographical film) (Evita, genre, musical), ...`}.

After generating triplet lists for the gold table and the model-generated answer table, we calculate the similarity between them using relative distance for numbers and Normalized Levenshtein Distance for text across each field in the triplet. Each triplet in the target is matched with its closest triplet in the prediction greedily and thus we can compute weighted precision and recall, in the same manner as Liu et al. (2023). The evaluation code will be released.

Model	simple			composition			intersection			all		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Oracle setting												
Gemini Pro	41.5	38.1	37.6	39.8	36.7	36.6	42.4	40.6	40.0	41.5	38.9	38.4
Gemini Flash	66.4	59.0	**60.4**	67.4	58.4	**60.4**	66.8	59.3	**61.1**	66.8	59.0	**60.7**
GPT4*o*	58.2	47.0	49.1	50.7	45.8	44.5	53.1	42.9	44.2	54.3	44.9	45.9
Human										81.2	74.0	**73.0**
Closed book setting												
PaLM-2	55.5	48.1	**49.6**	50.6	45.7	**46.6**	52.1	45.5	**46.6**	52.9	46.4	**47.6**
Gemma	38.9	26.7	27.9	37.4	25.8	26.6	39.3	26.2	27.7	38.8	26.3	27.5
Gemini Pro	46.9	30.3	31.1	43.8	29.5	30.4	47.8	32.0	33.5	46.6	30.9	32.0
Gemini Flash	37.4	29.0	31.3	36.2	27.2	29.6	36.6	28.8	30.7	36.8	28.5	30.7
GPT4*o*	46.8	26.6	29.1	28.1	16.0	18.1	35.9	22.3	24.4	37.9	22.4	24.6
Open book setting												
Tool LM	24.2	20.3	18.4	43.6	33.0	36.8	50.8	44.2	**46.6**	39.7	33.4	34.5

Table 5: Baseline performance by question type. For all question types, we observe Gemini Flash (60.7 F1) and PaLM-2 (47.6 F1) to outperform other baselines in oracle and closed book setting respectively, lagging 12.3 and 25.4 points behind the human baseline of 73.0.

5 Results & Discussion

In this section, we address key research questions: (1) Is TANQ a challenging dataset for state-of-the-art models? (2) How do models perform in a closed book setting compared to using external context (oracle)? (3) How effective are tool-augmented models on TANQ? (4) What challenges arise from different TANQ specifications—question types, reasoning skills, etc.? (5) What are the common failure cases of the evaluated models?

5.1 Question Types

In Table 5, we compare the performance of all oracle, close, and open book baselines. In the oracle setting, we find Gemini Flash consistently outperforming other models with an overall F1 score of 60.7, followed by GPT4*o*. However, still, a considerable gap remains to the **human baseline** of 73.0. Both models experience a significant drop in performance in the closed book setting, falling behind the smaller, best-performing PaLM-2 model. Notably, PaLM-2 achieves higher recall scores, indicating that while the other models can generate some rows of the answer tables, they are slightly less accurate than PaLM-2, and their resulting tables are shorter, containing fewer entities. Different to Gemini Flash/Pro and Gemma, GPT4*o* and PaLM-2 struggle with more complex questions types, i.e., composition and intersection questions. Moreover, the 9B-sized Gemma model demonstrates performance comparable to much larger models in closed book evaluation.

5.2 Evaluation with Diverse Prompts

In addition to the evaluations discussed in Section 5, we further assessed the baselines to examine the impact of (i) instruction tuning and (ii) demonstration selection on the models' performance on TANQ. Table 15 compares the performance of baselines across different numbers of demonstrations in the input prompt (i.e., 1-, 3-, and 5-shot), while Table 16 provides an overview of how different instruction styles affect the performance of Gemini Flash.

For almost all baselines, performance improves slightly when the number of input examples is increased from 1 to 3 in the prompt. However, no further improvements are observed with an increase to five examples, supporting our decision to evaluate models in a 3-shot setting. Except for Gemini Pro and GPT4*o*, manually selecting input examples does not yield enhanced performance.

When comparing the performance of the top-performing oracle baseline, Gemini Flash, across different instruction styles, we find that the detailed (B), step-by-step (C), and simple (D) instruction styles yield similar performance (see Figure 6). In contrast, using an empty instruction (A)—where we provide only input examples without additional details—results naturally in

Model	Filter num			Filter time			Filter entity			Date2Year			Quantity conv			Time calc			Approx		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Oracle setting																					
Gemini Pro	32.8	28.9	29.0	40.7	38.6	38.1	42.7	39.9	39.0	39.5	36.2	36.3	43.5	38.1	39.0	40.8	41.5	39.3	48.8	45.4	45.2
Gemini Flash	60.6	53.9	**55.3**	68.6	60.8	**62.6**	68.3	59.4	**61.6**	66.0	58.5	**60.1**	71.6	60.8	**63.5**	68.1	58.2	59.6	69.3	65.2	**65.5**
GPT4*o*	56.0	47.0	45.8	53.6	45.7	47.7	55.9	46.0	47.2	42.1	35.5	35.3	47.5	41.1	43.2	81.6	65.8	**71.8**	34.2	35.3	34.3
Closed book setting																					
PaLM-2	50.5	46.3	**46.9**	53.9	47.1	**48.0**	54.9	48.5	**49.7**	51.4	44.7	**46.1**	51.5	45.2	**46.8**	53.1	48.3	**50.1**	52.3	43.4	**45.7**
Gemma	37.4	26.5	27.8	37.1	26.4	27.3	40.3	26.2	27.6	40.4	27.4	28.7	37.4	23.8	25.2	39.2	27.4	29.3	40.8	27.5	29.6
Gemini Pro	50.2	28.3	29.3	48.5	31.4	32.4	46.3	31.2	32.4	46.1	30.6	31.9	48.0	31.4	33.5	51.4	33.9	35.3	42.8	30.2	31.4
Gemini Flash	38.7	29.5	32.1	35.8	26.7	29.1	36.2	27.7	29.9	35.9	27.8	29.7	36.6	28.0	29.1	34.8	27.5	30.1	37.1	27.1	30.0
GPT4*o*	30.7	21.8	24.5	37.8	24.0	24.9	46.5	25.6	28.7	31.2	18.9	20.6	25.5	12.7	16.1	48.3	25.3	26.6	24.0	17.6	19.6
Open book setting																					
Tool LM	48.7	39.1	43.1	36.1	29.7	30.3	37.6	33.0	34.2	53.1	37.8	41.9	33.8	28.4	30.2	25.8	18.2	21.1	48.3	44.0	45.9

Table 6: Baseline performance by skills required to answer the question: **Filter**ing with **num**erical/date**time/entity** conditions, date-to-year (**Date2Year**), quantity **conv**ersion, time **calc**ulation, and **approx**imation. In the oracle setting Gemini Flash performs best across all skills, while PaLM-2 performs better in the closed book evaluation.

decreased performance (see Table 16). For our evaluation in Section 5, we used the simple instruction style, as additional details as per instruction B and C did not contribute to observable performance improvements on the evaluated models.

5.3 Reasoning Skills

Table 6 presents the performance of models on more challenging questions requiring further skills such as filtering and time conversion.

Gemini. Overall, questions requiring filtering with numerical conditions pose the biggest challenge for Gemini models resulting in a performance drop compared to other skills. For example, Gemini Flash's performance drops from 60.7 F1 (overall) to 55.3 in oracle evaluation.

PaLM-2. For PaLM-2, we observe that the model struggles particularly with questions requiring numeracy, i.e., approximation of numerical values, quantity conversion, calculations based on datetime attributes, and filtering with numerical conditions. This challenge does not persist in the open book baseline where PaLM-2 is augmented with a calculator tool. Despite these limitations, the PaLM-2 model outperforms all other baselines across all skills in the closed book setting.

GPT4*o*. For GPT-4*o*, no clear patterns are observed regarding its limitations in specific reasoning skills. In both the oracle and closed book scenarios, the model struggles with date-to-year conversions and the approximation of numerical values. However, it outperforms other closed book baselines in time calculations.

Film	Duration	Genre	Publication
Evita	129 min	biographical film,musical	1996

Table 7: Example TANQ answer table with multiple entries in a single cell, i.e., ''biographical film, musical''.

Model	One skill			Two skills			Three skills		
	P	R	F1	P	R	F1	P	R	F1
Oracle setting									
Gemini Pro	42.3	39.9	39.3	36.3	32.4	32.4	46.5	41.0	41.3
Gemini Flash	66.8	58.6	**60.6**	67.4	59.4	**60.8**	74.5	68.8	**70.1**
GPT4*o*	51.5	42.4	43.3	50.7	46.4	47.6	51.0	39.7	44.7
Closed book setting									
PaLM-2	52.8	46.2	**47.4**	52.0	46.9	**47.9**	57.0	47.1	**50.5**
Gemma	37.9	25.8	26.9	42.4	28.9	30.3	42.5	26.2	29.5
Gemini Pro	46.9	30.4	31.5	46.4	32.8	34.4	50.1	30.2	31.0
Gemini Flash	35.6	27.8	29.9	37.4	28.3	30.6	39.5	23.2	25.9
GPT4*o*	39.4	24.7	26.5	32.5	14.5	17.6	9.4	1.8	3.0
Open book setting									
Tool LM	40.7	33.5	35.0	46.4	35.6	39.8	–	–	–

Table 8: Baseline performance by number of skills required to successfully answer the TANQ questions.

Number of Skills. Table 8 shows baseline performance by the number of skills required to generate the correct answer table. We observe across baselines a stable performance as the number of skills increases and questions become more complex. Except GPT4*o*, which shows significant performance decreases as the number of skills

Model	One Relation			Three Relations			Five Relations			Ten Relations		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Oracle setting												
Gemini Pro	41.6	39.2	38.6	42.2	40.7	39.5	38.5	36.2	35.0	35.8	36.0	35.8
Gemini Flash	67.0	59.5	**61.1**	68.5	59.2	**61.8**	64.7	58.6	**59.5**	64.7	58.4	**60.5**
GPT4*o*	55.1	44.7	45.8	63.2	53.6	53.5	74.9	43.3	50.2	23.8	19.4	20.5
Closed book setting												
PaLM-2	53.7	47.4	**48.4**	51.8	45.2	**46.0**	54.2	47.7	**48.8**	62.4	47.7	**48.9**
Gemma	38.5	26.3	27.6	35.6	25.6	26.5	47.6	30.5	31.1	42.0	35.8	37.8
Gemini Pro	45.6	32.6	33.5	47.7	30.8	31.4	51.1	29.4	29.9	43.9	39.2	40.9
Gemini Flash	36.6	28.6	30.7	37.7	28.4	31.0	35.3	26.9	28.6	36.1	26.1	29.4
GPT4*o*	37.5	21.9	24.4	33.6	25.1	27.8	55.6	15.4	18.2	34.3	21.3	23.2
Open book setting												
Tool LM	31.6	32.0	30.2	42.4	33.7	36.7	34.0	32.9	33.0	18.4	17.8	18.1

Table 9: Baseline performance by number of relations in TANQ questions (i.e., corresponds to number of columns in the answer table). Most baselines demonstrate a stable performance as relations increase.

increases in the close book setting, e.g., from 26.5 (one skill) to 17.6 (two skills).

5.4 Relations in TANQ Questions

Table 9 demonstrates baselines' performance across the number of relations in the given question. For example, in Figure 1 two relations, i.e., composer and release year, are requested for the movies specified in the question. While multiple models show a performance decrease with an increasing number of relations, this gap is particularly significant for the tool augmented model where the F1 score decrease by almost twelve points comparing questions requiring 1 relations vs. 10. Moreover, we observe for most baselines (oracle/close/open book) little performance drops comparing questions with 1 relation to those with 3—indicating that up to a certain limit models can successfully retrieve information for an increasing number of relations. However, the performance gaps increase for GPT4*o* (close) and the tool-augmented model as the number of relations increases further, i.e., from 3 to 5 and from 5 to 10.

5.5 Answer Table Length

Table 10 gives an overview of model performance across different answer table lengths: short answer tables (up to 3 rows), medium (up to 6 and long tables (7 or more rows). Models that perform well on longer tables are GPT-4*o*, Gemini Flash, and PaLM-2. As table size increases, so does the number of rows the models can correctly retrieve, resulting in higher recall scores, while precision slightly drops for most oracle baselines models. It is apparent that the model struggles with extracting correct information from the growing list of provided oracle documents as the table size increases. This is obviously not the case for close book evaluation as no oracle documents are given as input.

Model	Short			Medium			Long		
	P	R	F1	P	R	F1	P	R	F1
Oracle setting									
Gemini Pro	47.0	41.2	39.9	42.3	38.1	37.8	41.1	39.3	38.8
Gemini Flash	68.6	56.6	**58.3**	66.2	56.0	**58.0**	67.2	60.6	**62.2**
GPT4*o*	62.9	31.6	36.3	53.1	37.4	39.5	55.0	49.7	50.0
Closed book setting									
PaLM-2	56.4	43.9	**43.7**	54.5	45.0	**45.9**	51.9	47.3	**48.5**
Gemma	35.8	25.8	25.6	37.6	25.1	25.9	39.4	27.0	28.4
Gemini Pro	47.8	29.6	29.7	45.9	31.5	32.7	47.0	30.5	31.6
Gemini Flash	32.9	25.7	25.6	36.3	27.9	29.4	37.1	28.9	31.4
GPT4*o*	30.6	13.0	14.9	35.3	17.8	20.2	39.6	25.4	27.5
Open book setting									
Tool LM	23.2	29.2	23.0	33.3	31.7	30.9	43.3	34.3	36.5

Table 10: Baseline performance by length of answer tables. **Short** tables have <3 rows, **medium** tables up to six, and **long** tables >7 rows. Most baselines show little performance variation across table sizes.

5.6 Failures Types

Hence, we further study common failures cases of oracle baselines (see Table 11). The aim is to understand challenges related to TANQ when the model has access to necessary information, in form of oracle documents.

Failure Type	Gemini	PaLM-2	GPT4o	Tool LM
Rel missing	37.3	33.3	0	3.9
No header	27.5	3.9	0	5.9
Filter issues	15.7	15.8	4.0	5.9
Halluc. relations	5.9	9.8	5.9	2.0
Halluc. other	0	15.7	0	7.8
Missing entity	80.4	58.8	27.5	78.4
Non-table	31.4	21.6	0	15.7
Partial answers	3.9	23.5	2.0	0
Wrong answer	14.9	7.8	9.8	13.7

Table 11: Common failure types: **rel**ations **missing** in answer table, **no** column **header**, **filter** issues (e.g., filter condition ignored), **halluc**inated **relations**, **other** types of **halluc**inations, **missing entity** (i.e., row), output **not** in **table** format, **partial answers** in one or multiple table cells, **wrong answer** given in one or multiple cells. Scores given as % of all annotated samples.

We manually evaluate a subset of model predictions to identify common failure cases, categorized as follows: 1.) **Relation missing**: At least one expected relation (column) is missing; 2.) **No table header** generated; 3.) **Filtering**: Conditions (attribute, numeric, dates) ignored or incorrectly applied; 4.) **Hallucinated relations**: Unrequested columns are added; 5.) **Other hallucinations**; 6.) **Missing entity**: Expected rows are missing; 7.) **Non-table answers**: Text or non-table format returned; 8.) **Partial answers**: Incomplete information for a given entity or attribute; 9.) **Wrong answer (cell)**: Incorrect cell content.

Comparing open and oracle baselines, our observations show that the best performing model, w.r.t least failures, is GPT4o (oracle). The only significant failure category we identify for GPT4o are missing entities in generated answer tables. The most present issue we observe for Gemini Pro (oracle) are missing entities, missing relations, resulting in answer tables with a subset of columns, non-table outputs, and missing headers of generated tables. Similarly, multiple aspects pose a challenge for PaLM-2 (oracle), including hallucinations, missing entities/relations, and applying filtering conditions correctly. For the tool augmented model, missing entities is a frequent issue.

5.7 Detailed Evaluation of the Open Book Model

It is noteworthy that the open book, ReAct-based, baseline performed worse than other baselines, also the closed book ones. To better understand the limitations of the open book baseline, we manually evaluated the reasoning chains generated by the model for 50 TANQ samples. These chains outline the verbal reasoning processes and actions taken by the model in an interleaved manner. Our evaluation identified six different areas where the open book model struggles.

	name	located in
building	Old Main at the University of Arkansas	Arkansas

Table 12: Example issue related to formatting answer tables: The first column is redundant.

Retrieval of a Complete Entity List. The most prevalent issue, found in 19 out of 50 samples, was that the answer list did not contain all entities of the target table. Our review of the ReAct chain logs showed that the model often terminated the search for further entities after finding a list of entities. For example, it might start searching in the introduction section of a Wikipedia article and identify some movies directed by a particular director, but then fail to continue searching the rest of the page, missing additional films.

Generation of Correctly Formatted Answer Tables. We also found issues related to answer table formatting in 7 out of 50 samples. For instance, in one case, the name column was incorrectly replicated into a table row (i.e., ‘‘Building’’ below in Table 12).

Retrieval of Attributes. In six out of fifty samples, the model retrieved entities but failed to find attributes for all entities, resulting in partly empty rows (see Table 14).

Answering Intersection Questions. A similar issue arises with intersection questions, which ask for entities that fulfill multiple conditions. For example, ‘‘What science fiction films were both written and directed by Steven Spielberg? What is their genre, publication date, and duration?’’ The open book model often produced tables that fulfilled only one of these conditions, such as listing films written by Steven Spielberg without considering if he also directed them. This problem occurred in 5 out of the 50 evaluated examples.

Answering Composition Questions. As the name suggests, composition questions require the model to first retrieve a list of intermediate entities, which are then used to compile the final entity list

needed for the answer table. For example, consider the question: *''Who directed the film that Jules Feiffer wrote? What is their occupation, date of death, date of birth, country of citizenship, place of birth, and award received?''* The model should first retrieve the films as intermediate entities to identify the requested directors. However, in four out of the fifty evaluated samples, the model generated answer tables that included intermediate entities only instead of the requested entities. For instance, the answer table for the above question looked as demonstrated in Table 13. The first row correctly shows the directors, but the second and third rows contain the names of movies written by Jules Feiffer rather than their directors.

Applying Filters to the Generated Table. Finally, the model commonly ignored filtering conditions specified in the input question. We observed this in 10 out of 50 samples. For example, in the question: *''Who was a mayor of Saint Paul, MN, and what was their occupation, position held, and date of birth? Filter the answer table for date of birth equal to or before 1829,''* the resulting table included mayors born after 1829, failing to apply the requested filter.

6 Future Work

Our results on TANQ highlight the need for qualitative benchmarks, metrics, and modeling approaches to address current limitations.

We initially used the RMS metrics from Liu et al. (2023) to evaluate table generation. However, RMS was developed for the chart-to-table conversion task, which involves mostly numerical tables. In contrast, TANQ tables contain significant text content, leading to variations such as expressing ''USA'' as ''United States.'' Another difference is the more diverse table structure, where cells often contain lists of values. We adapted the RMS metrics for TANQ (see Section 5). Our experiments indicate that this version of RMS strikes a good balance of providing signal, being simple to explain and implement, and being fast to execute. However, further research on table evaluation is necessary to consider a broader range of formats.

Interestingly, we found that smaller models like Gemma (9B parameters) performed surprisingly well. While Gemma was not among the top models for any specific skill or question type in the closed book setting, it did not fall far behind models that are significantly larger, such as PaLM-2, Gemini Pro, and GPT-4*o*. This suggests that, despite its smaller size, models like Gemma and Gemini Flash offer promising results, though there is still room for improvement. While the size of Gemini Flash is undisclosed, it is known to be smaller than Gemini Pro, further highlighting that scale alone does not guarantee better performance on complex reasoning tasks. This raises an important question for future research: How can we further enhance the performance of smaller models?

Additionally, while tool-based LMs are popular, our results suggest that they may not be the optimal choice in domains where up-to-date knowledge is not required, especially for answering simple questions. For more complex questions involving intersecting information, tool-augmented LMs perform comparably to other baselines, but for simpler questions, they lag behind both oracle and closed book baselines.

Numeracy remains a challenge for state-of-the-art models. Despite advancements in numerical reasoning with LLMs, many models still struggle with questions requiring simple numerical skills such as filtering the content of a based on a numerical condition. While progress has been made in recent years (Imani et al., 2023; Akhtar et al., 2023; Chen et al., 2022), challenges remain.

Another challenge is that all evaluated models tend to generate incomplete answer tables, often missing some entities (rows) and relations (columns) from the target table. Future work should focus on developing methods to address these limitations and generate answer tables that capture the entire requested information. Addressing these issues and improving table evaluation metrics are important areas for future research.

7 Conclusion

This paper introduces TANQ, the first open domain question answering dataset where the answers require building tables from information across multiple sources. To create TANQ, we design and apply an automated dataset pipeline using large language models and Wikipedia and Wikidata as knowledge sources. We further release for each cell of the answer tables source attribution in form of text, tabulation, or infobox proofs. We evaluate our dataset on state-of-the-art models

in three different setting: oracle documents provided, closed, and open book setting. Our results and analysis suggest that TANQ is a complex task with many challenges ahead.

References

Mubashara Akhtar, Abhilash Shankarampeta, Vivek Gupta, Arpit Patil, Oana Cocarascu, and Elena Simperl. 2023. Exploring the numerical reasoning capabilities of language models: A comprehensive analysis on tabular data. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15391–15405, Singapore. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.findings-emnlp.1028

Samuel Joseph Amouyal, Ohad Rubin, Ori Yoran, Tomer Wolfson, Jonathan Herzig, and Jonathan Berant. 2022. QAMPARI: An open-domain question answering benchmark for questions with many answers from multiple paragraphs. *CoRR*, abs/2205.12665. https://doi.org/10.48550/ARXIV.2205.12665

Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy P. Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul Ronald Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, et al. 2023a. Gemini: A family of highly capable multimodal models. *CoRR*, abs/2312.11805. https://doi.org/10.48550/ARXIV.2312.11805

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernández Ábrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan A. Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vladimir Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, et al. 2023b. Palm 2 technical report. *CoRR*, abs/2305.10403. https://doi.org/10.48550/ARXIV.2305.10403

Ali Furkan Biten, Rubèn Tito, Andrés Mafla, Lluís Gómez, Marçal Rusiñol, Minesh Mathew, C. V. Jawahar, Ernest Valveny, and Dimosthenis Karatzas. 2019. ICDAR 2019 competition on scene text visual question answering. In *2019 International Conference on Document Analysis and Recognition, ICDAR 2019, Sydney, Australia, September 20–25, 2019*, pages 1563–1570. IEEE. https://doi.org/10.1109/ICDAR.2019.00251

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher 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 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6–12, 2020, virtual*.

Wenhu Chen, Ming-Wei Chang, Eva Schlinger, William Yang Wang, and William W. Cohen. 2021a. Open question answering over tables and text. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3–7, 2021*. OpenReview.net.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020a. Tabfact: A large-scale dataset for table-based fact verification. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26–30, 2020*. OpenReview.net.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020b. HybridQA: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1026–1036, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.findings-emnlp.91

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. 2021b. FinQA: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711, Online and Punta Cana, Dominican Republic, Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.emnlp-main.300

Zhiyu Chen, Shiyang Li, Charese Smiley, Zhiqiang Ma, Sameena Shah, and William Yang Wang. 2022. ConvFinQA: Exploring the chain of numerical reasoning in conversational finance question answering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6279–6292, Abu Dhabi, United Arab Emirates, Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.emnlp-main.421

Michael Chui, James Manyika, Jacques Bughin, Richard Dobbs, and Charles Roxburgh. 2012. The social economy: Unlocking value and productivity through social technologies.

Fredo Erxleben, Michael Günther, Markus Krötzsch, Julian Mendez, and Denny Vrandecic. 2014. Introducing wikidata to the linked data Web. In *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19–23, 2014. Proceedings, Part I*, volume 8796 of *Lecture Notes in Computer Science*, pages 50–65. Springer. https://doi.org/10.1007/978-3-319-11964-9_4

Vivek Gupta, Maitrey Mehta, Pegah Nokhiz, and Vivek Srikumar. 2020. INFOTABS: Inference on tables as semi-structured data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2309–2324, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.210

Rujun Han, Peng Qi, Yuhao Zhang, Lan Liu, Juliette Burger, William Yang Wang, Zhiheng Huang, Bing Xiang, and Dan Roth. 2023. RobustQA: Benchmarking the robustness of domain adaptation for open-domain question answering. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4294–4311, Toronto, Canada. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.findings-acl.263

Jonathan Herzig, Thomas Müller, Syrine Krichene, and Julian Eisenschlos. 2021. Open domain question answering over tables via dense retrieval. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 512–519, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.naacl-main.43

Shima Imani, Liang Du, and Harsh Shrivastava. 2023. MathPrompter: Mathematical reasoning using large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 37–42, Toronto, Canada. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.acl-industry.4

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1:*

Long Papers), pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics. https://doi.org/10.18653/v1/P17-1147

Amrith Krishna, Sebastian Riedel, and Andreas Vlachos. 2022. ProoFVer: Natural logic theorem proving for fact verification. *Transactions of the Association for Computational Linguistics*, 10:1013–1030. https://doi.org/10.1162/tacl_a_00503

Fangyu Liu, Julian Eisenschlos, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Wenhu Chen, Nigel Collier, and Yasemin Altun. 2023. DePlot: One-shot visual language reasoning by plot-to-table translation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10381–10399, Toronto, Canada. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.findings-acl.660

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, Dragomir Radev, and Dragomir Radev. 2022. FeTaQA: Free-form table question answering. *Transactions of the Association for Computational Linguistics*, 10:35–49. https://doi.org/10.1162/tacl_a_00446

Vaishali Pal, Andrew Yates, Evangelos Kanoulas, and Maarten de Rijke. 2023. MultiTabQA: Generating tabular answers for multi-table question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6322–6334, Toronto, Canada. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.acl-long.348

Morgane Rivière, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozinska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucinska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Juyeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjösund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, and Lilly McNealus. 2024. Gemma 2: Improving open language models at a practical size. *CoRR*, abs/2408.00118. https://doi.org/10.48550/arXiv.2408.00118

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal. Association for Computational Linguistics. https://doi.org/10.18653/v1/D15-1237

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics. https://doi.org/10.18653/v1/D18-1259

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1–5, 2023*. OpenReview.net.

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. https://doi.org/10.18653/v1/2022.acl-long.454

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. https://doi.org/10.18653/v1/2021.acl-long.254

A Modeling Details

For our experiments, we use the following GPT4*o* model available over the OpenAI API: `gpt-4o-2024-08-06`.[6] The PaLM-2 model we used for the pipeline and evaluation purposes was the model variant with 340B parameters.

B Prompt Templates

Figures 5 and 6 show the prompts used for evaluating the extracted evidence and rephrasing the TANQ questions to increase the naturalness of template-based questions. Figure 7 shows prompts for the four instruction variants we evaluated: empty, simple, detailed, and step-by-step.

[6] https://platform.openai.com/docs/models/gpt-4o.

Evidence Evaluation Prompt

Given a statement and a proof, decide if the statement is "verifiable" or "not verifiable". Only use the information provided in the proof and no background knowledge. A statement is not verifiable if the proof is missing some information. Reason step-by-step and then give a label for the prediction.

Examples:
Statement: Spartacus (film) director Stanley Kubrick.
Proof from Wikipedia page http://de.wikipedia.org/wiki/Stanley_Kubrick: Stanley Kubrick (July 26, 1928 - March 7, 1999) was an American film director, producer, screenwriter and photographer.
Output: While Stanley Kubrick was a film director, the proof doesn't mention if he directed Spartacus (film). Not verifiable.

Statement: Albert Einstein inventor theory of relativity.
Proof from Wikipedia page http://en.wikipedia.org/wiki/Albert_Einstein: Best known for developing the theory of relativity, he also made important contributions to quantum mechanics, and was thus a central figure in the revolutionary reshaping of the scientific understanding of nature that modern physics accomplished in the first decades of the twentieth century.
Output: The proof clearly mentions that Albert Einstein invented the theory of relativity. Verifiable.

Statement: Harmony Korine director Spring Breakers.
Proof from Wikipedia page http://en.wikipedia.org/wiki/Harmony_Korine: movie|original language|publication date|title|narrative location|country of origin|genre|distributed by
Spring Breakers|English|2013-03-19|Spring Breakers|Florida|United States of America|comedy film|A24
Trash Humpers|English|2009|Trash Humpers|Tennessee|United States of America|comedy horror|Drag City
Julien Donkey-Boy|English|1999|Julien Donkey-Boy|New York|United States of America|drama film|Fine Line Features
Mister Lonely|English|2007|Mister Lonely|Paris|France|comedy drama|IFC Films
The Beach Bum|English|2019-03-22|The Beach Bum|Florida|United States of America|comedy film|Neon
Gummo|English|1997|Gummo|Ohio|United States of America|teen film|Fine Line Features
Output: The proof mentions the movie Spring Breakers but not if Harmony Korine directed it. Not verifiable.

Figure 4: Prompt used for evidence evaluation. We prompt a language model to evaluate the extracted evidence in a natural language inference setting. The LLM labels the input statements as ‘‘verifiable’’ or ‘‘not verifiable’’ based on evidence provided in form of a sentence, table, or infobox.

Question Rephrasing Prompt

Rewrite the given question, following this instruction:
1. The rewritten question should sound natural.
2. If the question has errors (grammar, typos, etc.), correct them but don't change the meaning of the question.
3. The question has one or multiple words in parenthesis, for example "(distributed_by)". While rewriting keep the text in parenthesis unchanged!

Examples:
Question: Which film was directed and produced by Mel Brooks and what was their producer (producer), genre (genre), screenwriter (screenwriter),
country of origin (country_of_origin), distribution format (distribution_format), distributed by (distributed_by), cast member (cast_member), original language (original_language)?
Rewritten question: What is the genre (genre), country of origin (country_of_origin), distribution format (distribution_format) and original language (original_language) of films directed and produced by Mel Brooks? Moreover, who are their screenwriters (screenwriter), producer (producer), distributors (distributed_by), and cast members (cast_member)?

Question: Albert Uderzo illustrated what comic and what was their has author (has_author), publication date (publication_date), country of origin (country_of_origin), language (language), media franchise (media_franchise)?
Rewritten question: Albert Uderzo illustrated which comics and who were their authors (has_author)? When and in which countries (country_of_origin) were they published (publication_date)? Provide also information about the language (language) of the comics and the names of their media franchises (media_franchise).

Question: Who transferred to Pro Sesto in summer of 2020 who was an Italian footballer and what was their date of birth (date_of_birth), place of birth (place_of_birth), occupation (occupation)?
Rewritten question: Which Italian footballer transferred to Pro Sesto in summer of 2020 and what is their date of birth (date_of_birth), place of birth (place_of_birth), and occupation (occupation)?

Figure 5: Rephrasing questions increases the naturalness of template-based extension questions. To ensure the question meaning is preserved during rephrasing, we add structured annotations in parenthesis to questions with the name of each relation.

Name	Occupation	Date of Death	Date of Birth	Citizenship	Place of Birth	Award Received
Mike Nichols	director, producer	November 19, 2014	November 6, 1931	United States	Berlin, Germany	Academy Award [...]
Popeye	Sailor		January 17, 1929			
Munro						

Table 13: Example table for issues related to composition questions in open book baseline. While the question asks for directors, the second and third row contain movies.

	Director	Producer	Dir. of Photography	Composer	Genre	Publication
The Life and Death of Colonel Blimp	Michael Powell	E. Pressburger	Georges Périnal	Allan Gray	Drama, War	1943
A Canterbury Tale	Michael Powell	E. Pressburger	Erwin Hillier	Allan Gray	Drama, Romance	1944
A Matter of Life and Death	Jack Cardiff		Allan Gray		Fantasy, War	1946
The Tales of Hoffmann	Jacques Offenbach					

Table 14: Example table for issues related to missing attributes in answer tables. For the last row no relation other than director could be retrieved.

Model	**One Shot**			**Three Random Shots**			**Three Manual Shots**			**Five Shots**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Oracle setting												
Gemini Pro	41.8	38.2	37.8	41.5	38.9	38.4	50.8	40.0	39.0	41.8	38.2	37.8
Gemini Flash	54.7	56.8	50.8	66.8	59.0	60.7	67.1	57.7	59.4	54.7	58.2	53.3
GPT4*o*	55.4	44.9	46.6	54.3	44.9	45.9	39.9	34.8	35.4	48.8	40.1	40.7
Closed book setting												
PaLM-2	46.4	42.9	43.2	52.9	46.4	47.6	52.3	45.8	47.0	52.1	45.8	47.0
Gemma	28.0	18.2	19.2	38.8	26.3	27.5	40.9	30.6	31.8	38.5	33.1	34.6
Gemini Pro	37.9	33.6	34.3	46.6	30.9	32.0	44.9	26.8	30.5	39.2	28.7	29.9
Gemini Flash	21.3	25.9	18.6	36.8	28.5	30.7	33.5	30.2	30.1	29.9	28.6	23.9
GPT4*o*	37.2	23.6	26.3	37.9	22.4	24.6	66.1	15.7	16.2	38.1	18.9	20.4

Table 15: Evaluation results of models with different prompt settings. For most baselines the 3-shot setting yields the best results.

Model	**Empty Instruction**			**Detailed Instruction**			**Step-by-Step Instruction**			**Simple Instruction**		
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Oracle Setting	61.8	55.4	56.4	66.8	60.2	61.8	67.2	60.3	61.6	66.8	59.0	60.7
Closed Book Seeting	34.2	25.7	28.5	37.5	27.0	29.6	36.1	29.1	31.2	36.8	28.5	30.7

Table 16: Evaluation results of three-random-shots prompt with different instruction styles on Gemini Flash. We observe clear difference with/without instruction but almost the same result on different instruction styles. Figure 6 shows the text for the different instruction variants.

Question: Which indian movies were both directed and written by Kamal Haasan and who were their composers?

(A) 'Empty' Instruction

```
------------------
Examples: [...]
------------------
Question: {question}
Related Evidence:{proofs}
Answer table:
```

(B) Simple Instruction

```
Answer the given question by generating an answer table.
------------------
Examples: [...]
------------------
Question: {question}
Related Evidence:{proofs}
Answer table:
```

(C) Detailed Instruction

```
Answer the given question by generating an answer table.
Extract the table content from the provided evidence.
Using the evidence provided, extract only relevant
information to populate the table. Ensure that every cell
in the table is directly supported by the evidence.
------------------
Examples: [...]
------------------
Question: {question}
Related Evidence:{proofs}
Answer table:
```

(D) Step-by-Step Instruction

```
Answer the given question by generating an answer table.
Step 1: From the evidence provided, identify key entities
relevant to the question.
Step 2: Populate the first table column with the entities.
Step 3: From the evidence, extract all requested
attributes (e.g., date of birth, genre, or location).
Populate the remaining table cells with this information.
Step 4: Ensure that every cell in the table is directly
supported by the evidence.
------------------
Examples: [...]
------------------
Question: {question}
Related Evidence: {proofs}
Answer table:
```

Figure 6: Four instruction variants we evaluated: empty, simple, detailed, and step-by-step.

```
"init_qid": Question identifier,
"init_question": Original question from QAMPARI before adding attributes,
"ext_question": New question with added attributes,
"ext_question_cleaned": Question without attributes pareatheness,
"ext_question_rephrased": Question rephrased by a language model,
"question_properties": List of the added attributes into question,
"answer_list": List of rows answers
      "init_answer_wikidata_id": Answer ID in wikidata,
      "init_answer_wikipedia_id": Answer ID in wikipedia,
      "init_answer_composed":
      "extension_answer": List of properties (columns)
            "extension_property_id": Wikidata identifier for the property,
            "extension_property_label": Name for the property,
            "extension_entity": Dict of meta data of current extension answer,
            "extension_wikidata_id": Wikidata ID for the value of the property,
            "extension_wikipedia_id": Wikipedia ID for the value of the property,
            "proof": List of evidence for the property in Wikipedia documents,
      "init_answer_proof": List of initial proofs from QAMPARI
            "proof_text": Content of proof,
            "found_in_url": URL link where the proof is found,
      "filter_pass": Flag of whether this answer pass filters in question or not,
      "instance_type": Type of this answer, extracted from wikidata page,
"extended_types": List of successed extension types for question,
"answer_table": Golden answer table for the extended question.
```

Figure 7: Json schema description of a TANQ dataset entry outlining all components of a single dataset entry.