Questionnaire Meets LLM: A Benchmark and Empirical Study of Structural Skills for Understanding Questions and Responses

Duc-Hai Nguyen, Vijayakumar Nanjappan, Barry O'Sullivan, Hoang D. Nguyen*
Insight Research Ireland Centre for Data Analytics &
School of Computer Science and Information Technology,
University College Cork, Ireland
125109073@umail.ucc.ie, vnanjappan@ucc.ie, {b.osullivan, hn}@cs.ucc.ie
* Corresponding author: hn@cs.ucc.ie

Abstract

Millions of people take surveys every day, from market polls to medical questionnaires and customer feedback forms. These datasets capture valuable insights, but the ability of large language models (LLMs) to process questionnaire data, where lists of questions are crossed with hundreds of respondent rows, remains underexplored. Current survey analysis tools (e.g., Qualtrics, SPSS, REDCap) are designed for human operators, leaving practitioners without evidence-based guidance on how to best represent questionnaires for LLM consumption. We address this gap by introducing **QASU** (Questionnaire Analysis and Structural Understanding), a benchmark that probes six structural skills, including answer lookup, respondent count, and multi-hop inference, across six serialization formats and multiple prompt strategies. Experiments on five LLMs (GPT-5-mini, Gemini-2.5-Flash, Qwen3-32B, Llama3-70B, Amazon Nova Lite) show that format choice significantly impacts performance, with up to 9 percentage points improvement over baseline formats, and reveal substantial gaps (10 to 30 percentage points) between proprietary and open-weight models. Self-augmented prompting yields model-dependent benefits, proving effective for proprietary models but unreliable for open-weight alternatives. By systematically isolating format and prompting effects, our open-source benchmark offers practical guidance for advancing both research and real-world practice in LLM-based questionnaire analysis.

Keywords: Large Language Models, Questionnaire Data, Structural Understanding, Data Serialization, Prompt Engineering, Benchmarking, Information Retrieval

1. Introduction

Survey questionnaires collect responses from millions of people on a daily basis, whether it is a market poll, an academic study, a hospital satisfaction form, or a quick in-app feedback request (Goodfellow, 2023). This ongoing flow of answers provides essential insights that shape decisions in many areas. Market researchers use them to understand consumer preferences, healthcare teams track patient satisfaction to improve care quality, and educators rely on them to assess learning outcomes. As survey data accumulates at scale, the challenge is no longer just collecting it, but also making sense of it efficiently and accurately.

Questionnaires store structured data where each row records a respondent, each column a question, and each cell an answer. These structures enable fundamental operations like counting respondents, retrieving specific answers, or filtering by criteria (Groves et al., 2009; Gweon and Schonlau, 2024; Rink et al., 2024). Traditional tools for these operations (SPSS, REDCap, or bespoke SQL pipelines) require schema-specific configuration and programming expertise, placing them out of reach for many clinicians, educators, and social workers who collect questionnaire data but lack technical training. Integrating questionnaires with large language models (LLMs) could democratize access to these capabilities through natural-language interfaces, while also enabling tasks beyond traditional analysis: generating summaries, identifying patterns, and reasoning over complex answer distributions. Before deploying LLMs in such workflows, however, we need systematic evidence on *where* and *how* they fail on structured questionnaire data.

Recent advances in LLMs have shown promise on table reasoning benchmarks like TabFact (Chen et al., 2020) and tasks like MMLU (Hendrycks et al., 2021) through chain-of-thought reasoning (Wei et al., 2022). However, most prior work assumes clean, database-like tables with uniform schemas and resolved categorical values (Herzig et al., 2020; Yin et al., 2020; Chen et al., 2021). Questionnaires retain structural complexities absent from these benchmarks: mixed answer types (multiple choice, Likert scales, free text), grouped questions, and skip logic. QASU evaluates LLM performance on questionnaire data in its raw state, where models must handle heterogeneous data types, resolve answer codes via schema metadata, and interpret varying response sets.

Before reasoning, questionnaires must be serialized into linear text. Serialization practices vary widely: social sciences use XML-based DDI Codebook (Vardigan et al., 2008), linked-data projects use JSON-LD or Turtle (Scandolari et al., 2021), healthcare uses HL7 FHIR JSON (International,

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 4728–4746
11–16 May 2026.

2019), and applied researchers use CSV/TSV (Gweon and Schonlau, 2024; Rink et al., 2024). This fragmentation creates uncertainty about which format best supports LLM reasoning over questionnaire data.

To address these challenges, we introduce **QASU**[1]: a benchmark that evaluates LLM performance on six micro-tasks, from counting respondents to reverse look-ups, across six serialization styles (HTML, XML, JSON, Markdown, plain text, and Turtle). Our results reveal three trends: (1) Open-weight models struggle on basic structural operations, with accuracy as low as 45% on respondent counting tasks for certain format combinations. (2) Format choice significantly impacts performance: HTML achieves up to 9 percentage points higher accuracy than TTL on answer lookup tasks. (3) Self-augmented prompting (Sui et al., 2024) yields model-dependent benefits: effective for proprietary models (up to +20pp on some datasets) but unreliable for open-weight alternatives, which uniformly fail to benefit from self-augmentation.

These findings have direct implications for medical research, social science, and business analytics, where effective serialization and prompting strategies enable practitioners to integrate LLMs into questionnaire analysis workflows.

Our contributions are:

- QASU, the first benchmark to isolate multiple structural skills of LLMs on questionnaire data.
- We provide a complete ablation of serialization and prompt options, turning what is now guesswork into practical guidance (see § 4).
- Empirical evidence that self-augmented prompting (Sui et al., 2024) yields model-dependent gains on questionnaire reasoning, with practical implications for selecting appropriate prompting strategies based on model architecture.

2. Preliminaries

2.1. Questionnaire Data Structure

Questionnaire data typically appear as flat respondent-question matrices where rows represent respondents and columns represent questions (Groves et al., 2009). More complex instruments introduce skip logic, multi-select grids, or nested sections (Couper, 2000; Dillman et al., 2014). We focus primarily on flat formats for their prevalence and comparability.

Cell contents vary widely: plain text for open-ended responses, numeric values for Likert scales, categorical labels for multiple-choice, and date/time fields. The coexistence of heterogeneous data types creates a representational gap between the two-dimensional grid and the linear token sequence expected by LLMs (Fowler, 2013; Herzig et al., 2020; Yin et al., 2020).

2.2. Questionnaire Serialization and Splitting

Serialization converts two-dimensional questionnaires into linear sequences for LLM input. Practices vary by domain: social sciences use XML-based DDI Codebook (Vardigan et al., 2008), linked-data projects use JSON-LD or Turtle (Scandolari et al., 2021), healthcare uses HL7 FHIR JSON (International, 2019), and applied research uses TSV or Markdown (Gweon and Schonlau, 2024; Rink et al., 2024). QASU evaluates six formats under identical conditions.

For long questionnaires exceeding context limits, we randomly sample respondents while preserving space for one-shot examples, following (Brown et al., 2020). This avoids positional bias and maintains consistent prompt structure.

3. QASU Benchmark

QASU systematically examines how input design affects LLM performance on questionnaire data. Prior studies applying LLMs to questionnaires (Gweon and Schonlau, 2024; Rink et al., 2024) differ substantially in input preparation (CSV vs. Markdown), making results hard to compare. QASU addresses this by crossing six serialization formats with five prompt configurations across six micro-tasks drawn from routine survey-analysis workflows. Details appear in § 3.3.

3.1. Structural Understanding Capabilities

We categorize the fundamental abilities required to work effectively with questionnaire data into two complementary dimensions, illustrated in Figure 1. These dimensions parallel the way human analysts approach data: first locating relevant information, then performing computations or logical reasoning over it.

1) Search & Retrieval. The first capability is finding information accurately in specific cells, rows, or columns of questionnaire data. This is similar to table QA, where even a simple value lookup means matching the question to the right parts of the table and handling complex structures (Herzig et al., 2020). In questionnaires, this means interpreting column headers, matching query terms to the appropriate variables, and identifying the correct answers without losing the context of the sur-

[1] https://github.com/ReML-AI/QASU

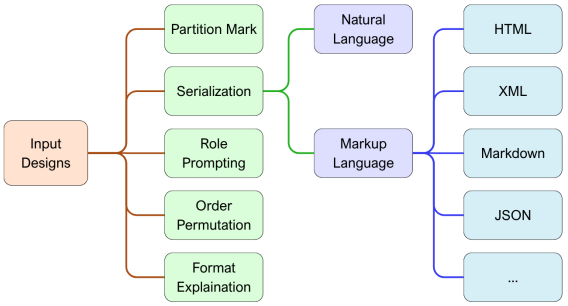

Figure 1: Input designs evaluated in the QASU benchmark. Each design is a combination of serialization format, layout choices, and prompt annotations.

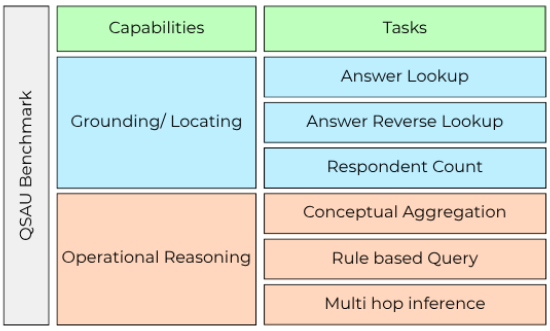

Figure 2: Task types in the QASU benchmark. Each task is designed to evaluate a specific structural skill in processing questionnaire data.

rounding data. Diverse serialization formats, such as CSV, JSON, and XML, encode structure in very different ways, which can influence how easily an LLM can parse and navigate the data. A robust retrieval capability is therefore foundational: without it, downstream reasoning steps, no matter how sophisticated, cannot proceed reliably. QASU isolates this skill to assess how different input designs affect an LLM's ability to recognize and extract the correct information across varying formats.

2) Aggregation & Reasoning. Besides locating specific entries, questionnaire analysis often needs to combine multiple pieces of information, apply filters, and perform computations to provide answers for more complex queries. This includes tasks where models must integrate several constraints, group related concepts, or apply logical and numerical operations across subsets of the data. Such operations are common in both human-led survey analysis and structured benchmarks like Spider (Yu et al., 2018), and they require the model to maintain a consistent mapping between the schema, the constraints, and the relevant rows. Crucially, these reasoning processes depend on accurate retrieval as an initial step; an error in locating the relevant data propagates directly into incorrect aggregate results.

3.2. Task Design

The six tasks in QASU represent fundamental operations behind questionnaire analysis: retrieving answers, filtering by criteria, counting records, and combining conditions. These form the building blocks for more complex reasoning such as generating reports or identifying statistical patterns. Our focus on foundational operations enables systematic assessment of structural understanding before advancing to specialized capabilities such as missing-data imputation, temporal dynamics, or hierarchical structures, which introduce orthogonal complexity dimensions. Although some tasks superficially resemble database QA benchmarks such as Spider (Yu et al., 2018) or WikiSQL (Zhong et al., 2017), questionnaires introduce distinct challenges: rows are organized around individual respondents rather than entity-relation schemas; answer cells mix heterogeneous types (Likert scores, free text, coded multiple-choice) within the same table; and answer codes must be resolved to human-readable labels via schema metadata, an operation absent from clean SQL tables where categorical values are already resolved.

From this foundation, QASU evaluates structural understanding over serialized questionnaires rather than database tables. Let a serialized document be $D = (Q, R)$, where Q maps question keys to their descriptors (for example, the text and option set for "Medical Condition"), and $R = \{r_i\}$ is a set of respondent records. Each record r_i contains an identifier and an `answers` map from question keys to atomic values or codes. The concrete representation of D varies by format. In JSON it is a nested object with keys and values, in HTML it is a structured markup fragment, in Markdown it appears as a text table, and in XML it is a tag hierarchy. Across formats, the model must align a natural language query to a question key and then resolve the correct value or record by following the appropriate path in D. An overview of benchmark tasks is illustrated in Figure 2.

Answer Lookup. Given a respondent identifier i and a target question key k, return the value v such that $r_i.\texttt{answers}[k] = v$. This tests schema grounding and path resolution across formats. For JSON, this is a direct key access; for HTML or Markdown, it requires locating the field by header text and reading the associated entry in the respondent's record.

Reverse Answer Lookup. Given a question key k and a target value v that may be encoded as a letter for multiple choice, return the set of respondent identifiers $\{i : r_i.\texttt{answers}[k] = v\}$. This checks whether the model can search records by field value and, when applicable, map codes to their option

Table 1: Simple query templates for the six structural tasks evaluated in the QASU benchmark.

Task	Input
Answer Lookup	What is the [attribute] of [respondent]? Provide concisely only the attribute of the respondent
Answer Reverse Lookup	Which respondent has [attribute]? Provide concisely only the respondent number. Use "," to separate if there are more than 1 respondent
Respondent Count	How many respondents are there? Provide concisely only the number of respondents
Conceptual Aggregation	How many respondents have [attribute]? Provide concisely only the number of respondents have that attribute
Rule-based Querying	Find all respondents matching [criteria]? Provide concisely only the respondent number. Use "," to separate if there are more than 1 respondent
Multi-hop relational inference	Which respondent has [attribute_1] and [attribute_2]? Provide concisely only the respondent number. Use "," to separate if there are more than 1 respondent

labels using the information in Q.

Respondent Count. Return the number of respondent records $|R|$, or the number that satisfy a stated global criterion when present. This requires detecting respondent boundaries in the serialized document and avoids double counting when formatting introduces repeated headers or decorative text.

Conceptual Aggregation. Given a high-level concept defined by a set of labels under a single question key, count the respondents whose answer falls in that set. For example, with $k =$ "Medical Condition" and a concept defined as {*Diabetes*, *Obesity*}, return the total number of records whose $\texttt{answers}[k]$ matches any member of the set, with code-to-label decoding when needed.

Multi-hop Relational Inference. Find the respondent IDs that meet all given conditions for different questions. This means matching values across fields and taking only the records that satisfy every condition. The task gets harder when there are more conditions and when answer codes need to be converted to their full text.

Rule-based Querying. Given a numerical predicate over a field, return the identifiers of respondents that satisfy the predicate. For example, with $k =$ "Billing Amount" and a threshold, the model must parse numbers from text, apply the comparator, and report the matching records.

The tasks progress from direct retrieval to constrained reasoning. This ordering allows us to study how format and prompt choices affect alignment to question keys, robustness of path resolution across different serializations, and the reliability of composition over multiple constraints. Table 1 lists the prompt templates used for each task.

3.3. Data Collection and Reformatting

Datasets. We use five publicly available datasets spanning health (Prasad, 2020), psychology (Topp and Thomas, 2015), HCI (Group, 2022), software engineering (Overflow, 2022), and medical education. Each is reformatted into a unified representation serializable into JSON, HTML, XML, TTL, TXT, and Markdown. We randomly sample respondents to fit token budgets while keeping all questions intact.

Leakage-resistant. To prevent data leakage, we apply systematic obfuscation (Magar and Schwartz, 2022; Thakkar et al., 2021): rank swapping (Domingo-Ferrer and Torra, 2001) (5% window) for numeric variables and 10% probabilistic perturbation (Warner, 1965) for categorical variables. Perturbation rates preserve first- and second-order statistics (Drechsler et al., 2011), ensuring observed performance differences reflect genuine understanding.

3.4. Evaluation

We evaluate using exact-match accuracy after normalizing whitespace and case, following WikiSQL (Zhong et al., 2017) and TabFact (Chen et al., 2020). Exact match is appropriate here because all six QASU tasks have deterministic ground-truth answers: lookups return specific cell values, counts return integers, and set-valued tasks (Reverse Lookup, Multi-hop, Rule-based Query) return exact respondent ID lists. Each prompt includes an explicit output-format instruction requiring human-readable values and decoded option labels, and over 90% of model outputs adhere to these constraints. While complementary metrics such as token-level F_1 or relaxed set matching could provide finer-grained signal for partial-credit scenarios, exact match offers a strict, unambiguous baseline that avoids interpretive discretion and enables direct comparison across formats and models. The benchmark tests six formats (JSON, XML, Markdown, HTML, Turtle, TXT) and four prompt augmentations: (1) syntax description, (2) partition markers, (3) role labels (Liu et al., 2022), and (4) format explanations. This yields a rich input design space (Figure 1), with results in Table 2.

3.5. Self-Augmented Prompting

Following (Sui et al., 2024), we adopt *self-augmented prompting* adapted for questionnaire data. This two-stage approach (Figure 3) first prompts the model to generate structural probes (respondent ID patterns, question headers, value aliases), then incorporates this output into the task prompt. This anchors reasoning in explicit structural knowledge, relying entirely on the LLM's own

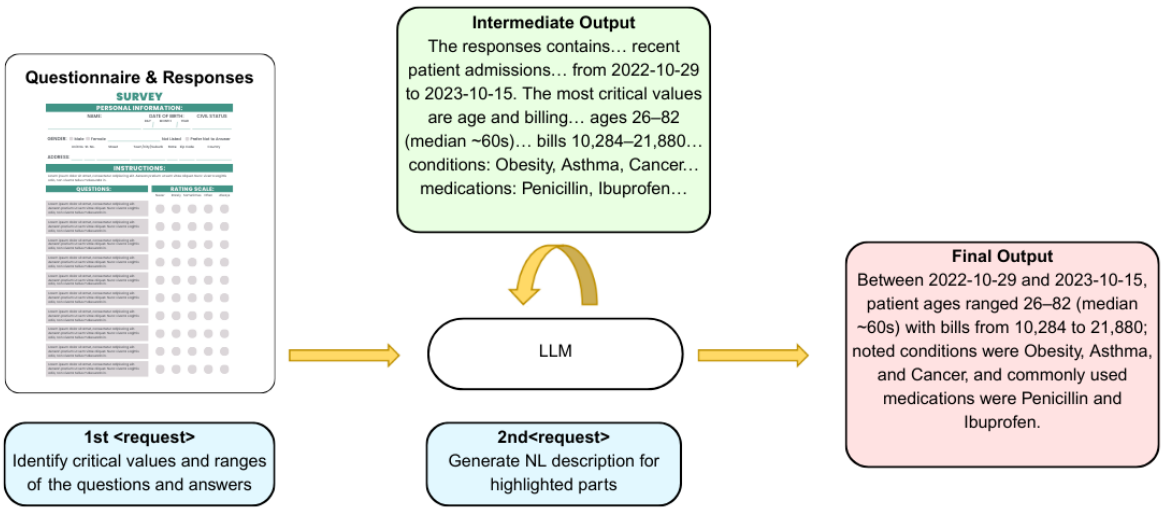

Figure 3: Illustration of self-augmented prompting. This process consists of two phases: 1) using self-augmented prompts to ask the LLM to generate additional knowledge (intermediate output) about the table; 2) incorporating the self-augmented response into the second prompt to request the final answer for a downstream task. As depicted in the figure, the LLM is able to identify important values in the table, which assists in generating a more accurate answer for the downstream task.

parsing capabilities.

4. Experiments

4.1. Experimental Settings

Models. We evaluate five large language models: two proprietary models (**GPT-5-mini**, **Gemini-2.5-Flash**) and three open-weight alternatives (**Qwen3-32B**, **Llama3-70B**, **Amazon Nova Lite**). Unless otherwise stated, all completions are generated with temperature set to 0 and one output per prompt. This setup follows the deterministic setting used in earlier research on table reasoning and text-to-SQL (Brown et al., 2020), which means that any changes in performance are due to how the input was designed, not to random sampling. By setting the decoding parameters, we can directly link changes in accuracy to structural prompts or serialization format, not to randomness in the creation process.

Datasets and query pool. To examine whether the structural capabilities measured by QASU generalize across domains, we compile a diverse evaluation set from five publicly available questionnaire datasets covering health, psychology, software engineering, and clinical training. These include: (1) an anonymized hospital discharge dataset from **Kaggle's Healthcare Dataset** (Prasad, 2020); (2) **Self-reported mental well-being responses** hosted on Mendeley Data (Topp and Thomas, 2015); (3) the **SUS-UTA7** usability study for wearable systems (Group, 2022); (4) the **2022 Stack Overflow Developer Survey** (Overflow, 2022); and (5) an **ISBAR** dataset rating medical students' patient hand-off performance.

For each dataset, we generate 245 independent benchmark cases per QASU task, where each case is a random sample of respondents fitted to the token budget, paired with one of the six task templates from Table 1. This produces a balanced total of 1,470 evaluated cases per model (each case is serialized into 6 different data formats). Each dataset's question list contains mixed types of questionnaire data such as Likert scales, multiple-choice grids, and open-ended text responses, preventing a model from memorizing or exploiting a single questionnaire schema.

Evaluation metric. We check the correctness of all outputs by comparing them to the ground truth via exact match, calculated after normalizing whitespace and case. This decision is consistent with established protocols in structured-data reasoning benchmarks like WikiSQL and TabFact (Zhong et al., 2017; Chen et al., 2020), where deterministic ground-truth answers eliminate the need for semantic similarity scoring. To reduce confusion caused by the format, each prompt includes an explicit output-format instruction emphasizing that answers must be human-readable values. It also specifies that letter codes from multiple-choice questions must be matched to their corresponding option text. This constraint, along with deterministic decoding, makes it possible to compare serialization formats and prompting variants in a clear and repeatable way.

4.2. Results

Cross-Model Performance. Proprietary models (GPT-5-mini, Gemini-2.5-Flash) outperform open-weight alternatives (Qwen3-32B, Llama3-70B, Nova Lite) by 10–30pp on most tasks (Table 2). The gap is task-dependent: proprietary models achieve 86–96% on Respondent Count and Concept Aggregation versus 59–93% for open-weight models. Multi-hop Relational Inference is hardest for all models (31–41% open-weight, 66–78% proprietary), indicating that compositional reasoning over multiple constraints exposes fundamental limitations.

Format Sensitivity. Format preferences diverge markedly across models. GPT-5-mini favors HTML and Markdown for complex reasoning (Answer Lookup: 75.4% HTML vs. 66.6% TTL), while Gemini-2.5-Flash performs best with plain text on several tasks (TXT: 88.0% Answer Lookup, 78.3% Multi-hop). Open-weight models show less consistent format preferences. Interestingly, structured markup formats (JSON, XML, HTML) do not universally outperform plain text, suggesting that narrative-style serialization may reduce parsing overhead for certain architectures.

Input Design Ablations. Table 3 isolates prompt component contributions. Partition marks emerge as most critical across all models: removing them degrades GPT-5-mini performance by 8–33pp depending on task, with Multi-hop suffering the largest drop (16.6pp). Open-weight models exhibit even stronger reliance, with Qwen3-32B losing 32.1pp on Rule-based Query and Nova Lite dropping 25.4pp on Answer Lookup. Counter-intuitively, Llama3-70B improves when removing one-shot examples on Answer Lookup (+8.7pp) and Concept Aggregation (+10.3pp), suggesting examples may introduce confusion for certain architectures.

Error Pattern Analysis. The quantitative results reveal four systematic failure patterns. *(i) Respondent boundary confusion*: Respondent Count accuracy drops sharply for open-weight models in formats that lack explicit record delimiters: Qwen3-32B falls from 70.6% (Markdown, where pipe-delimited rows mark boundaries) to 59.7% (TTL, where records are separated only by blank nodes). This pattern suggests models miscount when structural cues are implicit. *(ii) Answer code resolution failure*: Reverse Lookup consistently underperforms Answer Lookup across all models (Table 2), with gaps of 3–14pp for proprietary models and 13–24pp for open-weight models. Since Reverse Lookup requires mapping coded values (e.g., "B") back to option text using schema metadata, this gap isolates code-to-label resolution as a distinct bottleneck. *(iii) Multi-constraint composition breakdown*: Multi-hop accuracy is the lowest across all models (31–41% open-weight, 66–78% proprietary), even though its constituent operations (single-field lookup, filtering) are individually well-handled. The compounding of errors across conditions explains the disproportionate difficulty. *(iv) Format-dependent schema misalignment*: XML produces the best Answer Lookup for Llama3-70B (81.1%) but the worst Reverse Lookup (50.0%), a 31pp swing between tasks that use the same format. This indicates that the model aligns to XML tag hierarchies for forward navigation but struggles to invert the traversal direction, an asymmetry absent in flatter formats like Markdown.

Self-Augmented Prompting. Table 5 compares three self-augmented prompting variants against standard 1-shot baselines. Results are highly model-dependent: Gemini-2.5-Flash achieves dramatic gains on Healthcare (+20pp) and ISBAR (+9pp) when generating critical values, yet performance collapses on SO-2022 and SUS-UTA7 (25–28pp drops). Open-weight models uniformly fail to benefit from self-augmentation, with self critical values identification causing 6–10pp drops below 1-shot. We attribute this divergence to differences in stage-1 probe quality: self-augmentation requires the model to first generate accurate structural summaries (respondent ID patterns, question headers, value ranges), and errors in these intermediate outputs propagate to stage-2 reasoning. The consistent underperformance of open-weight models on basic structural tasks (Table 2) suggests their stage-1 probes are less reliable, compounding rather than correcting downstream errors. For practitioners, this means self-augmentation should be reserved for models that already demonstrate strong baseline structural understanding; otherwise, simpler prompt augmentations (partition marks, role labels) remain more practical and robust.

5. Related Work

LLMs and structured data. Large language models have been increasingly applied to tasks involving structured inputs like databases, forms, and semi-structured text. Earlier approaches combined natural-language understanding with symbolic execution, helping to translate questions into SQL queries or logical forms (Zhong et al., 2017; Yu et al., 2018). Later approaches improved results by pretraining models on large collections of structured data (Herzig et al., 2020; Liu et al., 2022).

Other researchers have created artificial datasets to help models handle a wider variety of data formats (Yu et al., 2021), set up extra

Table 2: Comprehensive benchmark results for all tasks. 'GPT' refers to GPT-5-mini, 'Gemini' to Gemini-2.5-Flash, while 'Qwen3', 'Llama3', and 'Nova' correspond to Qwen3-32B, Llama3-70B, and Amazon Nova Lite, respectively. Each cell's color is scaled relative to other formats within the same task and model, where deeper colors signify superior performance.

	Answer Lookup					Reverse Lookup				
Format	GPT	Gemini	Qwen3	Llama3	Nova	GPT	Gemini	Qwen3	Llama3	Nova
HTML	**75.4%**	86.9%	67.6%	64.6%	57.1%	**88.0%**	**80.6%**	50.1%	52.7%	45.3%
JSON	72.0%	84.6%	75.4%	80.6%	57.7%	86.9%	80.0%	51.7%	57.4%	48.4%
MD	72.0%	84.6%	73.1%	78.3%	62.3%	86.3%	78.3%	**55.5%**	**63.1%**	**50.7%**
TTL	66.6%	86.3%	72.6%	71.4%	58.9%	85.1%	76.6%	54.6%	57.3%	47.3%
TXT	70.9%	**88.0%**	**78.9%**	74.3%	**62.9%**	85.7%	80.0%	55.1%	55.1%	49.9%
XML	72.0%	77.7%	70.9%	**81.1%**	57.1%	86.3%	78.3%	53.9%	50.0%	43.9%

	Resp. Count					Concept. Agg				
Format	GPT	Gemini	Qwen3	Llama3	Nova	GPT	Gemini	Qwen3	Llama3	Nova
HTML	94.9%	93.7%	65.1%	72.6%	63.7%	**96.6%**	94.9%	64.1%	61.1%	64.9%
JSON	**96.0%**	94.9%	68.3%	80.6%	66.3%	96.0%	**96.0%**	72.1%	69.7%	**70.4%**
MD	94.9%	**96.0%**	**70.6%**	**93.1%**	**68.3%**	95.4%	93.7%	**74.9%**	82.3%	68.6%
TTL	94.3%	95.4%	59.7%	85.7%	58.4%	**96.6%**	95.4%	63.6%	73.1%	67.4%
TXT	94.3%	**96.0%**	69.7%	84.0%	64.9%	95.4%	93.1%	71.6%	77.7%	**70.4%**
XML	95.4%	93.1%	61.6%	89.1%	63.0%	95.4%	94.3%	65.9%	**86.3%**	70.3%

	Rule-based Query					Multi-hop				
Format	GPT	Gemini	Qwen3	Llama3	Nova	GPT	Gemini	Qwen3	Llama3	Nova
HTML	94.3%	91.4%	69.0%	54.3%	61.6%	**72.6%**	71.4%	33.3%	36.9%	32.1%
JSON	92.6%	92.0%	**72.7%**	59.3%	**67.9%**	66.3%	73.7%	32.1%	33.3%	30.4%
MD	93.7%	92.6%	69.6%	64.9%	66.7%	72.0%	75.4%	34.7%	38.3%	36.0%
TTL	**95.4%**	**93.7%**	72.3%	65.6%	62.2%	67.4%	74.3%	31.0%	35.7%	33.1%
TXT	94.3%	90.9%	70.6%	**78.4%**	**67.9%**	69.1%	**78.3%**	**35.6%**	**41.4%**	**36.3%**
XML	93.7%	88.6%	69.0%	62.0%	64.0%	68.6%	68.0%	34.4%	38.3%	31.0%

rules during the answer-generation process to ensure the outputs are always valid (Scholak et al., 2021), and mixed structured data with regular text to improve how well models can reason (Jiang et al., 2023). Recent studies also show that careful prompt design can boost model performance, even in situations where the model has not seen similar tasks before. Research such as Zero-Shot Chain-of-Thought and Table Meets LLM demonstrates that careful wording, formatting, and use of delimiters can make LLMs much better at working with structured data (Kojima et al., 2022; Sui et al., 2024). Our benchmark builds on these findings but focuses on questionnaires, which often have more complex relationships within each record and a wider variety of question types than typical database tasks.

Prompt engineering and self-augmentation. Recent prompting strategies, such as chain-of-thought reasoning (Wei et al., 2022), self-consistency (Wang et al., 2023), and least-to-most prompting (Zhou et al., 2023), encourage models to articulate their reasoning step by step or decompose complex problems into simpler sub-questions. Building on these ideas, Sui et al. introduced *self-augmented prompting*, where the model first generates explicit structural hints and then uses them to answer the question (Sui et al., 2024). We adapt this approach to questionnaire grids and observe model-dependent improvements over direct prompting, without adding any handcrafted logic. Complementary work on benchmark design for language resources emphasizes the importance of systematic evaluation protocols and controlled data splits to ensure reproducibility (Gorman and Bedrick, 2019), principles we adopt in QASU's construction.

Questionnaire Standards and Interoperability. There is no single agreed-upon standard for representing questionnaires, which creates challenges for both sharing and automated analysis. The DDI Codebook is common in social-science archives (Vardigan et al., 2008), FHIR JSON is required for many clinical forms, and the Survey Ontology encodes forms in linked data triples (Scandolari et al., 2021). A recent study of German health research identified thirty-two semantic standards and seven syntactic standards in active use (Vorisek et al., 2024). Our results show that this lack of consistency has a measurable impact: for example, HTML with embedded format notes achieves six percentage points higher accuracy than raw JSON, likely because it conveys richer structural information to the model. Similar challenges with format heterogeneity have been documented in work on multilingual resource standardization (Bel et al., 2016) and

Table 3: Study of different input design choices against the HTML baseline. 'GPT' refers to GPT-5-mini, 'Gemini' to Gemini-2.5-Flash, 'Qwen3' to Qwen3-32B, 'Llama3' to Llama3-70B, and 'Nova' to Amazon Nova Lite.

	Answer Lookup		Reverse Lookup		Resp. Count		Concept. Agg.		Rule-based Query		Multi-hop	
Input Design	Acc	Δ	Acc	Δ	Acc	Δ	Acc	Δ	Acc	Δ	Acc	Δ
GPT-5-mini												
Markup Lang. HTML	75.4%	0.0%	88.0%	0.0%	94.9%	0.0%	96.6%	0.0%	94.3%	0.0%	72.6%	0.0%
w/o format explanation	74.3%	-1.1%	84.0%	-4.0%	92.6%	-2.3%	93.1%	-3.5%	87.4%	-6.9%	72.0%	-0.6%
w/o partition mark	66.9%	-8.5%	72.0%	-16.0%	86.3%	-8.6%	89.1%	-7.5%	80.0%	-14.3%	56.0%	-16.6%
w/o role prompting	75.4%	0.0%	79.4%	-8.6%	90.3%	-4.6%	92.6%	-4.0%	81.7%	-12.6%	69.9%	-5.7%
w/o change order	64.0%	-11.4%	60.0%	-28.0%	81.1%	-13.8%	77.7%	-18.9%	64.0%	-30.3%	49.1%	-23.5%
w/o 1-shot	65.1%	-10.3%	77.1%	-10.9%	88.6%	-6.3%	88.6%	-8.0%	79.4%	-14.9%	47.4%	-25.2%
Gemini-2.5-Flash												
Markup Lang. HTML	86.9%	0.0%	80.6%	0.0%	93.7%	0.0%	94.9%	0.0%	91.4%	0.0%	71.4%	0.0%
w/o format explanation	81.7%	-5.2%	80.0%	-0.6%	93.2%	-0.5%	94.9%	0.0%	91.1%	-0.3%	74.3%	+2.9%
w/o partition mark	62.3%	-24.6%	60.6%	-20.0%	75.4%	-18.3%	71.1%	-23.8%	75.7%	-15.7%	57.1%	-14.3%
w/o role prompting	82.9%	-4.0%	81.1%	+0.5%	92.0%	-1.7%	96.0%	+1.1%	86.9%	-4.5%	71.2%	-0.2%
w/o change order	82.9%	-4.0%	74.9%	-5.7%	96.6%	+2.9%	93.7%	-1.2%	82.3%	-9.1%	68.6%	-2.8%
w/o 1-shot	74.9%	-12.0%	83.4%	+2.8%	96.0%	+2.3%	93.1%	-1.8%	90.9%	-0.5%	65.1%	-6.3%
Qwen3-32b												
Markup Lang. HTML	60.6%	0.0%	50.1%	0.0%	65.1%	0.0%	64.1%	0.0%	69.0%	0.0%	33.3%	0.0%
w/o format explanation	60.6%	0.0%	31.4%	-18.7%	37.7%	-27.4%	42.9%	-21.2%	36.9%	-32.1%	21.1%	-12.2%
w/o partition mark	76.6%	+9.0%	43.1%	-7.0%	66.3%	+1.2%	64.0%	-0.1%	50.0%	-19.0%	28.3%	-5.0%
w/o role prompting	70.9%	+3.3%	44.3%	-5.8%	52.9%	-12.2%	52.0%	-12.1%	43.4%	-25.6%	27.1%	-6.2%
w/o change order	72.0%	+4.4%	48.0%	-2.1%	53.1%	-12.0%	58.3%	-5.8%	37.7%	-31.3%	45.1%	+11.8%
w/o 1-shot	66.3%	+1.3%	34.4%	-15.7%	49.1%	-16.0%	56.0%	-8.1%	34.3%	-34.7%	21.4%	-11.9%
Llama3-70b												
Markup Lang. HTML	77.6%	0.0%	52.7%	0.0%	85.6%	0.0%	81.1%	0.0%	54.3%	0.0%	36.9%	0.0%
w/o format explanation	79.4%	+1.8%	41.1%	-11.6%	80.0%	-5.6%	79.4%	-1.7%	37.7%	-16.6%	21.7%	-15.2%
w/o partition mark	78.3%	+0.7%	40.6%	-12.1%	82.9%	-2.7%	82.9%	+1.8%	51.7%	-2.6%	35.1%	-1.8%
w/o role prompting	83.4%	+5.8%	40.0%	-12.7%	96.0%	+10.4%	95.4%	+14.3%	34.6%	-19.7%	22.6%	-14.3%
w/o change order	74.0%	-3.6%	42.9%	-9.8%	82.1%	-3.5%	89.6%	+8.5%	31.4%	-22.9%	25.6%	-11.3%
w/o 1-shot	86.3%	+8.7%	40.0%	-12.7%	90.9%	+5.3%	91.4%	+10.3%	48.6%	-5.7%	31.4%	-5.5%
Amazon Nova Lite												
Markup Lang. HTML	67.1%	0.0%	45.3%	0.0%	63.7%	0.0%	64.9%	0.0%	61.6%	0.0%	32.1%	0.0%
w/o format explanation	56.0%	-11.1%	35.7%	-9.6%	61.6%	-2.1%	52.0%	-12.9%	52.1%	-9.5%	24.9%	-7.2%
w/o partition mark	41.7%	-25.4%	28.0%	-17.3%	47.4%	-16.3%	44.3%	-20.6%	46.6%	-15.0%	24.9%	-7.2%
w/o role prompting	56.6%	-10.5%	34.0%	-11.3%	57.1%	-6.6%	50.3%	-14.6%	52.9%	-8.7%	23.4%	-8.7%
w/o change order	67.8%	+0.7%	37.1%	-8.2%	58.9%	-4.8%	72.1%	+7.2%	62.0%	+0.4%	23.7%	-8.4%
w/o 1-shot	63.1%	-4.0%	36.9%	-8.4%	48.5%	-15.2%	66.3%	+1.4%	49.1%	-12.5%	22.4%	-9.7%

Table 4: Self-augmented prompting instructions.

Method	**Instruction**
Format explanation	Generate short format specification and description of the survey within five sentences.
Critical values and ranges identification	Identify critical values and ranges of the survey related within five sentences.
Structural information description	Describe structural information, patterns and statistics of the survey within five sentences.

metadata annotation for structured corpora (Ide and Pustejovsky, 2017), and structure-aware generation continues to be explored in applied domains (Chen et al., 2025).

6. Conclusion

QASU is the first benchmark for probing LLM structural understanding of questionnaire data. By systematically varying serialization formats and prompt augmentations across six tasks, we show that format choice significantly impacts performance (up to 10pp accuracy gaps), with partition marks and role labels providing the most consistent gains. Self-augmented prompting (Sui et al., 2024) yields model-dependent benefits: effective for proprietary models but unreliable for open-weight alternatives. Our findings reveal substantial performance gaps between proprietary and open-weight models (10–30pp on most tasks), indicating that current open alternatives lack robust schema understanding for questionnaire data. These results provide actionable guidance for practitioners integrating LLMs into survey analysis workflows across healthcare,

Table 5: Comparison of self-augmented and 1-shot prompting across various design variants on downstream tasks. Refer to Table 4 for details on different self-augmented prompting options. "SA" stands for "Self-augmented"

Type	Choice	Healthcare	ISBAR	Mental-health	SO-2022	SUS-UTA7
		Acc	Acc	Acc	Acc	Acc
GPT 5 Mini						
1-shot	1-shot	45.0%	68.0%	**85.0%**	**84.0%**	**95.0%**
1-shot	w/o change order	47.0%	60.0%	62.0%	66.0%	85.0%
1-shot	w/o format explanation	**50.0%**	67.0%	78.0%	81.0%	84.0%
1-shot	w/o partition mark	46.0%	70.0%	74.0%	75.0%	83.0%
1-shot	w/o role prompting	48.0%	68.0%	80.0%	79.0%	88.0%
SA	self format explanation	49.0%	**72.0%**	72.0%	70.0%	73.0%
SA	self critical values and ranges identification	30.0%	71.0%	57.0%	58.0%	77.0%
SA	self structural information description	42.0%	**72.0%**	68.0%	66.0%	70.0%
Gemini 2.5 Flash						
1-shot	1-shot	50.0%	82.0%	**95.0%**	**83.0%**	91.0%
1-shot	w/o change order	42.0%	70.0%	89.0%	79.0%	90.0%
1-shot	w/o format explanation	45.0%	73.0%	89.0%	**83.0%**	**94.0%**
1-shot	w/o partition mark	44.0%	63.0%	80.0%	74.0%	90.0%
1-shot	w/o role prompting	40.0%	70.0%	92.0%	80.0%	92.0%
SA	self format explanation	35.0%	83.0%	90.0%	82.0%	63.0%
SA	self critical values and ranges identification	**70.0%**	**91.0%**	58.0%	58.0%	60.0%
SA	self structural information description	45.0%	78.0%	77.0%	72.0%	58.0%
Qwen3-32b						
1-shot	1-shot	**38.0%**	**57.0%**	**70.0%**	**65.0%**	**76.0%**
1-shot	w/o change order	32.0%	50.0%	62.0%	57.0%	68.0%
1-shot	w/o format explanation	34.0%	52.0%	65.0%	59.0%	70.0%
1-shot	w/o partition mark	36.0%	54.0%	67.0%	61.0%	72.0%
1-shot	w/o role prompting	33.0%	51.0%	63.0%	58.0%	69.0%
SA	self format explanation	37.0%	56.0%	69.0%	64.0%	75.0%
SA	self critical values and ranges identification	30.0%	46.0%	59.0%	53.0%	66.0%
SA	self structural information description	35.0%	53.0%	66.0%	60.0%	72.0%
Llama3-70b						
1-shot	1-shot	**42.0%**	**61.0%**	**75.0%**	**69.0%**	**79.0%**
1-shot	w/o change order	37.0%	54.0%	68.0%	62.0%	72.0%
1-shot	w/o format explanation	39.0%	56.0%	70.0%	64.0%	74.0%
1-shot	w/o partition mark	41.0%	58.0%	72.0%	66.0%	76.0%
1-shot	w/o role prompting	38.0%	55.0%	68.0%	63.0%	73.0%
SA	self format explanation	40.0%	60.0%	73.0%	68.0%	77.0%
SA	self critical values and ranges identification	34.0%	49.0%	64.0%	57.0%	69.0%
SA	self structural information description	39.0%	57.0%	71.0%	65.0%	75.0%
Amazon Nova Lite						
1-shot	1-shot	**33.0%**	**55.0%**	66.0%	**62.0%**	**74.0%**
1-shot	w/o change order	28.0%	48.0%	60.0%	55.0%	66.0%
1-shot	w/o format explanation	30.0%	50.0%	63.0%	57.0%	68.0%
1-shot	w/o partition mark	32.0%	52.0%	65.0%	59.0%	70.0%
1-shot	w/o role prompting	29.0%	49.0%	61.0%	56.0%	67.0%
SA	self format explanation	31.0%	54.0%	**67.0%**	61.0%	72.0%
SA	self critical values and ranges identification	26.0%	44.0%	56.0%	50.0%	64.0%
SA	self structural information description	30.0%	51.0%	64.0%	58.0%	70.0%

social science, and business analytics. For researchers developing future models, QASU isolates specific structural weaknesses that must be addressed to achieve human-level questionnaire comprehension. Future work should extend QASU to multilingual settings, where additional challenges arise from non-Latin answer codes, culturally specific Likert anchors (e.g., differing agreement scales across cultures), and surveys containing metadata in multiple languages within a single instrument. More complex reasoning patterns, including temporal dynamics and missing-data handling, also remain open directions.

7. Limitations

Several limitations should be noted when interpreting our results. First, QASU currently evaluates only English-language questionnaires. Multilingual extension faces non-trivial challenges, including culturally variable codebooks, non-Latin metadata fields, and language-specific response norms that may alter both serialization and prompt design. Second, our evaluation protocol excludes null and missing values from the reasoning target. In real-world survey analysis, missingness patterns are often informative (e.g., systematic non-response to sensitive questions), and models should ideally handle rather than ignore them. Third, QASU focuses on flat respondent–question matrices. Real-world instruments frequently employ skip logic, nested sections, or branching paths, which introduce structural complexity beyond the scope of the current benchmark. Fourth, we rely exclusively on exact-match accuracy. While appropriate for the deterministic tasks in QASU (see § 3.4), complementary metrics such as token-level F_1 or relaxed set matching could provide richer signal for set-valued outputs where partial credit is meaningful. Finally, our error analysis is based on aggregate performance patterns across tasks and formats rather than systematic inspection of individual model outputs. Future work should incorporate fine-grained output annotation to identify specific failure modes at the instance level.

8. Ethical Considerations

Questionnaire data frequently contains sensitive or personally identifiable information, such as health records, demographic profiles, or workplace feedback. Even when identifiers are removed, indirect re-identification remains possible through unique combinations of answers. To mitigate these risks, all source datasets in QASU are drawn from public releases that have been anonymized by their original curators, and we discard entries with excessive missing values to avoid leaking potentially identifying patterns in response sparsity. No manual inspection of raw identifiable content is performed beyond automated schema validation.

A further consideration is the cultural and linguistic scope of our benchmark. The current QASU release focuses exclusively on English-language questionnaires, most of which originate from Western healthcare, usability, and software-engineering contexts. Extending QASU to cover multilingual or cross-cultural settings would require additional safeguards, including culturally sensitive schema interpretation and bias detection. Finally, the evaluation protocol assumes that all null or missing values in the data are to be excluded from the reasoning target. This guarantees fairness in assessing structural comprehension; yet, it contrasts with certain real-world survey analysis situations where managing missing data is essential to the task. Researchers applying QASU insights to downstream systems should carefully reassess this assumption in operational contexts where missing values may themselves be informative.

We believe that responsible benchmarking on questionnaire data requires not only technical rigor in measuring model capabilities, but also careful attention to privacy, cultural representativeness, and task realism. Addressing these aspects is essential for any future expansion of QASU toward a broader and more diverse set of structured-data reasoning challenges.

9. Acknowledgements

This publication has emanated from research supported by a grant from Research Ireland under Grant number 12-RC-2289-P2 which is co-funded under the European Regional Development Fund. For the purpose of Open Access, the author has applied a CC BY public copyright license to any Author Accepted Manuscript version arising from this submission.

10. Bibliographical References

Nuria Bel, Sergio Espeja, and Montserrat Marimon. 2016. Abstracts in multiple languages: a multilingual and cross-lingual resource. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3815–3820.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, et al. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33*, pages 1877–1901.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Y. Wang. 2020. Tabfact: A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations*.

Yuyan Chen, Qingpei Guo, Shuangjie You, and Zhixu Li. 2025. Medtranstab: Advancing medical cross-table tabular data generation. In *Proceedings of the 18th ACM International Conference on Web Search and Data Mining (WSDM '25)*, New York, NY, USA. ACM.

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. 2021. 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. Association for Computational Linguistics.

Mick P. Couper. 2000. Web surveys: A review of issues and approaches. *Public Opinion Quarterly*, 64(4):464–494.

Don A. Dillman, Jolene D. Smyth, and Leah Melani Christian. 2014. *Internet, Phone, Mail, and Mixed-Mode Surveys: The Tailored Design Method*, 4th edition. John Wiley & Sons.

Josep Domingo-Ferrer and Vicenç Torra. 2001. Rank swapping for partial confidentiality. In *Proceedings of the First International Conference on Privacy in Statistical Databases*, pages 177–188. Springer.

Jörg Drechsler, Stefan Bender, and Susanne Rässler. 2011. Empirical evaluation of cross-sectional and longitudinal record linkage methods. *Journal of Official Statistics*, 27(3):403–421.

Floyd J. Fowler. 2013. *Survey Research Methods*, 5th edition. SAGE Publications.

Lynda T Goodfellow. 2023. An overview of survey research. *Respiratory Care*, 68(9):1309–1313.

Kyle Gorman and Steven Bedrick. 2019. We need to talk about standard splits. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2786–2791.

MIMBCD-UI Group. 2022. Sus-uta7: Usability scores for wearable-technology studies. https://github.com/MIMBCD-UI/dataset-uta7-sus.

Robert M. Groves, Floyd J. Fowler, Mick P. Couper, James M. Lepkowski, Eleanor Singer, and Roger Tourangeau. 2009. *Survey Methodology*, 2 edition. John Wiley and Sons.

Hyukjun Gweon and Matthias Schonlau. 2024. Automated classification for open-ended questions with bert. *Journal of Survey Statistics and Methodology*, 12(2):493–504.

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

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Mueller, Francesco Piccinno, and Julian Martin 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. ACL.

Nancy Ide and James Pustejovsky. 2017. Representing linguistic data: The text encoding initiative. In Nancy Ide and James Pustejovsky, editors, *Handbook of Linguistic Annotation*, pages 1289–1316. Springer.

HL7 International. 2019. Fhir structured data capture: Questionnaire resource. https://hl7.org/fhir/questionnaire.html.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Xin Zhao, and Ji-Rong Wen. 2023. Structgpt: A general framework for large language model to reason over structured data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9237–9251, Singapore. Association for Computational Linguistics.

Takeshi Kojima, Shixiang Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*, 35:22199–22213.

Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. 2022. Tapex: Table pre-training via learning a neural sql executor. In *International Conference on Learning Representations*.

Inbal Magar and Roy Schwartz. 2022. Data contamination: From memorization to exploitation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 157–165, Dublin, Ireland. Association for Computational Linguistics.

Stack Overflow. 2022. Developer survey 2022. https://survey.stackoverflow.co/2022/.

Prasad Prasad. 2020. Healthcare dataset on kaggle. https://www.kaggle.com/datasets/prasad22/healthcare-dataset.

Lois Rink, Job Meijdam, and David Graus. 2024. Aspect-based sentiment analysis for open-ended hr survey responses. In *Proceedings of the First Workshop on NLP for Human Resources*, pages 16–26, Malta. Association for Computational Linguistics.

Damiano Scandolari, Gloria Re Calegari, Irene Celino, and Mario Scrocca. 2021. The survey ontology (v1.1). Technical report, Cefriel. Available at https://cefriel.github.io/survey-ontology/.

Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. 2021. Picard: Parsing incrementally for constrained auto-regressive decoding from language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9895–9901. Association for Computational Linguistics.

Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. 2024. Table meets llm: Can large language models understand structured table data? a benchmark and empirical study. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining (WSDM '24)*, pages 645–654, New York, NY, USA. ACM.

Om Thakkar, Swaroop Ramaswamy, Rajiv Mathews, and Françoise Beaufays. 2021. Understanding unintended memorization in federated learning. In *Proceedings of the Third Workshop on Privacy in Natural Language Processing*, pages 1–10. Association for Computational Linguistics.

Stephanie Topp and Becky Thomas. 2015. Self-reported mental health survey. https://data.mendeley.com/datasets/bytb22nf7m/1.

Mary Vardigan, Pascal Heus, and Wendy Thomas. 2008. Data documentation initiative: Toward a standard for the social sciences. *International Journal of Digital Curation*, 3(1):107–134.

Carina Nina Vorisek, Sophie Anne Inès Klopfenstein, Matthias Löbe, Carsten Oliver Schmidt, Paula Josephine Mayer, Martin Golebiewski, and Sylvia Thun. 2024. Towards an interoperability landscape for a national research data infrastructure for personal health data. *Scientific Data*, 11(1):772.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain-of-thought reasoning in language models. In *International Conference on Learning Representations*.

Stanley L. Warner. 1965. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35*, pages 24824–24837.

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. Tabert: Pretraining for joint understanding of textual and tabular data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8413–8426. ACL.

Tao Yu, Chien-Sheng Wu, Xi Victoria Lin, Bailin Wang, Yi Chern Tan, Xinyi Yang, Dragomir Radev, Richard Socher, and Caiming Xiong. 2021. Grappa: Grammar-augmented pre-training for table semantic parsing. In *International Conference on Learning Representations*.

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. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921, Brussels, Belgium. Association for Computational Linguistics.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2sql: Generating structured queries from natural language using reinforcement learning.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V. Le, and Ed H. Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. In *International Conference on Learning Representations*.

A. Appendix - Prompt Templates

This appendix presents the prompt templates and example model responses used in the QASU benchmark. All task types share the same base structure; only the `<task>` content differs per query.

A.1. Base Prompt Structure

Each prompt sent to the LLM consists of seven XML-delimited sections. Figure 4 shows the complete base prompt template with all sections: `<example>`, `<questionnaire>`, `<role>`, `<format>`, `<output>`, and `<task>`.

A.2. Ablation Variants

To isolate the contribution of each prompt component, we create five ablation variants by removing or modifying one section at a time:

- **w/o role prompting**: removes the entire `<role>` section.
- **w/o format explanation**: removes the entire `<format>` section.
- **w/o one-shot example**: removes the entire `<example>` section.
- **w/o partition marks**: strips all XML tags (`<role>`, `<format>`, etc.) but retains their content.
- **w/o change order**: moves the `<questionnaire>` section from second position to the end, after `<task>`.

A.3. Self-Augmented Prompt Templates

Self-augmented prompting adds an `<approach>` section before `<output>` that instructs the model to generate intermediate structural knowledge, along with an `<output_format>` section that enforces a two-stage `### ANALYSIS ###` / `### ANSWER ###` response structure. Figures 5–7 show the three augmentation variants.

A.4. Example Model Responses

Figures 8–10 show example responses from GPT-5-mini on an answer lookup task using the healthcare dataset in JSON format, demonstrating the two-stage analysis-then-answer output produced by each self-augmented prompting variant.

Base Prompt Template

```
<example>
[One-shot example: questionnaire data + question + expected answer]
</example>

<questionnaire>
[Serialized questionnaire data in one of six formats]
</questionnaire>

<role>
You are a meticulous data analyst AI. Your primary function is to accurately analyze structured data and
provide precise, verifiable answers.
</role>

<format>
You will be given a dataset with two key parts:

A 'questions' section: This is your data dictionary or schema. It is the single source of truth for
understanding what each field and score means. Refer to it carefully.

A 'responses' section: This contains the raw data from each individual respondent.

To answer the question correctly, you must first use the 'questions' schema to fully understand the context
and meaning of the data points within the 'responses'. Do not rely on any prior knowledge outside of this
provided data. Base your entire analysis on the information given.
</format>

<output>
Provide your final answer directly and concisely. Output must be the human-readable value, not an option
code.

Rules:
- For MCQ fields, map any letter codes in the Responses (case-insensitive) to the corresponding option text
from the Questions schema, and output only that text. Do NOT include the letter or both.
  Example: If the schema is "[MCQ: A. Abnormal B. Inconclusive C. Normal]" and the response is "a",
output "Abnormal".
- For open-ended fields, output the exact text value.
- If the question asks for a count, provide only the number (e.g., "42").
- If the question asks for a list of names or IDs, provide a simple comma-separated list (e.g., "17, 21, 23").
- Preserve capitalization exactly as in the schema (e.g., output "Abnormal", not "abnormal" or "A").
</output>

<task>
[Natural-language question]
</task>
```

Figure 4: Base prompt template used across all QASU benchmark tasks. The template consists of seven XML-delimited sections providing one-shot examples, questionnaire data, role definition, format explanation, output rules, and the task question.

Self-Augmented Prompt Template: Format Explanation

<example>
[One-shot example: questionnaire data + question + expected answer]
</example>

<questionnaire>
[Serialized questionnaire data]
</questionnaire>

<role>
You are a meticulous data analyst AI. Your primary function is to accurately analyze structured data and provide precise, verifiable answers.
</role>

<format>
You will be given a dataset with two key parts:

A 'questions' section: This is your data dictionary or schema. It is the single source of truth for understanding what each field and score means. Refer to it carefully.

A 'responses' section: This contains the raw data from each individual respondent.

To answer the question correctly, you must first use the 'questions' schema to fully understand the context and meaning of the data points within the 'responses'. Do not rely on any prior knowledge outside of this provided data. Base your entire analysis on the information given.
</format>

<approach>
Generate short format specification and description of the survey within five sentences.
</approach>

<output_format>
Structure your response:

ANALYSIS
[Your analysis here]

ANSWER
[Your final answer]

END
</output_format>

<output>
Provide your final answer directly and concisely. Output must be the human-readable value, not an option code.

Rules:
- For MCQ fields, map any letter codes in the Responses (case-insensitive) to the corresponding option text from the Questions schema, and output only that text. Do NOT include the letter or both.
  Example: If the schema is "[MCQ: A. Abnormal B. Inconclusive C. Normal]" and the response is "a", output "Abnormal".
- For open-ended fields, output the exact text value.
- If the question asks for a count, provide only the number (e.g., "42").
- If the question asks for a list of names or IDs, provide a simple comma-separated list (e.g., "17, 21, 23").
- Preserve capitalization exactly as in the schema (e.g., output "Abnormal", not "abnormal" or "A").
</output>

<task>
[Natural-language question]
</task>

Figure 5: Self-augmented prompt template with format explanation augmentation. The `<approach>` section instructs the model to first generate a format specification and description of the survey before answering.

Self-Augmented Prompt Template: Critical Values

<example>
[One-shot example: questionnaire data + question + expected answer]
</example>

<questionnaire>
[Serialized questionnaire data]
</questionnaire>

<role>
You are a meticulous data analyst AI. Your primary function is to accurately analyze structured data and provide precise, verifiable answers.
</role>

<format>
You will be given a dataset with two key parts:

A 'questions' section: This is your data dictionary or schema. It is the single source of truth for understanding what each field and score means. Refer to it carefully.

A 'responses' section: This contains the raw data from each individual respondent.

To answer the question correctly, you must first use the 'questions' schema to fully understand the context and meaning of the data points within the 'responses'. Do not rely on any prior knowledge outside of this provided data. Base your entire analysis on the information given.
</format>

<approach>
Identify critical values and ranges of the survey related within five sentences.
</approach>

<output_format>
Structure your response:

ANALYSIS
[Your analysis here]

ANSWER
[Your final answer]

END
</output_format>

<output>
Provide your final answer directly and concisely. Output must be the human-readable value, not an option code.

Rules:
- For MCQ fields, map any letter codes in the Responses (case-insensitive) to the corresponding option text from the Questions schema, and output only that text. Do NOT include the letter or both.
  Example: If the schema is "[MCQ: A. Abnormal B. Inconclusive C. Normal]" and the response is "a", output "Abnormal".
- For open-ended fields, output the exact text value.
- If the question asks for a count, provide only the number (e.g., "42").
- If the question asks for a list of names or IDs, provide a simple comma-separated list (e.g., "17, 21, 23").
- Preserve capitalization exactly as in the schema (e.g., output "Abnormal", not "abnormal" or "A").
</output>

<task>
[Natural-language question]
</task>

Figure 6: Self-augmented prompt template with critical values augmentation. The `<approach>` section instructs the model to identify critical values and ranges in the survey data.

Self-Augmented Prompt Template: Structural Information

<example>
[One-shot example: questionnaire data + question + expected answer]
</example>

<questionnaire>
[Serialized questionnaire data]
</questionnaire>

<role>
You are a meticulous data analyst AI. Your primary function is to accurately analyze structured data and provide precise, verifiable answers.
</role>

<format>
You will be given a dataset with two key parts:

A 'questions' section: This is your data dictionary or schema. It is the single source of truth for understanding what each field and score means. Refer to it carefully.

A 'responses' section: This contains the raw data from each individual respondent.

To answer the question correctly, you must first use the 'questions' schema to fully understand the context and meaning of the data points within the 'responses'. Do not rely on any prior knowledge outside of this provided data. Base your entire analysis on the information given.
</format>

<approach>
Describe structural information, patterns and statistics of the survey within five sentences.
</approach>

<output_format>
Structure your response:

ANALYSIS
[Your analysis here]

ANSWER
[Your final answer]

END
</output_format>

<output>
Provide your final answer directly and concisely. Output must be the human-readable value, not an option code.

Rules:
- For MCQ fields, map any letter codes in the Responses (case-insensitive) to the corresponding option text from the Questions schema, and output only that text. Do NOT include the letter or both.
  Example: If the schema is "[MCQ: A. Abnormal B. Inconclusive C. Normal]" and the response is "a", output "Abnormal".
- For open-ended fields, output the exact text value.
- If the question asks for a count, provide only the number (e.g., "42").
- If the question asks for a list of names or IDs, provide a simple comma-separated list (e.g., "17, 21, 23").
- Preserve capitalization exactly as in the schema (e.g., output "Abnormal", not "abnormal" or "A").
</output>

<task>
[Natural-language question]
</task>

Figure 7: Self-augmented prompt template with structural information augmentation. The `<approach>` section instructs the model to describe structural patterns and statistics of the survey.

Example LLM Response for Self-Augmented Prompting (Format Explanation)

ANALYSIS
The dataset is structured as a JSON object with two top-level keys: "questions" and "responses". The "questions" section serves as the schema, defining 14 fields including both open-ended fields (Age, Date of Admission, Billing Amount, Room Number, Discharge Date) and multiple-choice fields (Gender, Blood Type, Medical Condition, Insurance Provider, Admission Type, Medication, Test Results). The "responses" section contains an array of 6 respondent records, each identified by a unique "respondent" ID and an "answers" map. Multiple-choice answers are stored as single letter codes (e.g., "B" for Male) that must be resolved against the schema. Open-ended fields store their values directly as strings.

ANSWER
19747

END

Figure 8: Example model response for self-augmented prompting with format explanation. The model first generates a format specification describing the dataset structure before producing the final answer.

Example LLM Response for Self-Augmented Prompting (Critical Values)

ANALYSIS
The survey contains 6 respondents with IDs: 109, 9, 8, 24, 56, 60. Ages range from 23 to 73 years old. Billing amounts span from 3909 to 37735, with respondent 60 having the highest billing amount and respondent 8 having the lowest. Admission types include Elective (A), Emergency (B), and Urgent (C), with respondent 56 having an Urgent admission. The date range of admissions covers 2020-01-21 to 2024-03-24 across the dataset.

ANSWER
19747

END

Figure 9: Example model response for self-augmented prompting with critical values. The model first identifies key values and ranges in the data before answering.

Example LLM Response for Self-Augmented Prompting (Structural Information)

ANALYSIS
The dataset contains 14 question fields and 6 respondent records in a flat respondent-question matrix. Of the 14 fields, 5 are open-ended (Age, Date of Admission, Billing Amount, Room Number, Discharge Date) and 9 are multiple-choice with coded letter responses. The gender distribution shows 5 males (code B) and 1 female (code A). Medical conditions represented include Asthma (2 respondents), Hypertension (2 respondents), Diabetes (1 respondent), and Obesity (1 respondent). Each respondent record follows an identical structure with a "respondent" identifier and an "answers" object mapping all 14 fields.

ANSWER
19747

END

Figure 10: Example model response for self-augmented prompting with structural information. The model first describes structural patterns and statistics before answering.