Tabular Data in Interactive and Conversational AI: A Survey of Foundations, Benchmarks, Systems, and Open Problems

Anonymous authors
Paper under double-blind review

Abstract

Tabular and structured data underlie much of modern analytical work, yet natural language systems for interacting with such data have largely been studied in fragmented subfields. This survey studies that landscape under the broader problem of *conversational AI over tabular and structured data*: systems that support multi-turn, context-dependent interaction with tables, databases, spreadsheets, and hybrid table–text documents. We first clarify the problem setting by defining tabular data, conversational interaction, and the primary interaction modes that distinguish querying, translating, manipulating, and orchestrating over structured data, while treating exploration as a recurrent interaction pattern rather than a separate category. Using an explicit corpus-construction and evidence policy, we organize 106 unique cited works into five categories: *Foundations*, *Conversational Table Question Answering* (CTabQA), *Conversational Text-to-SQL* (CText2SQL), *Interactive Table Manipulation*, and *Agentic Table Systems*. Across these categories, we compare benchmark datasets, modelling paradigms, and evaluation practices, while tracing how closely related problems have often been studied under different task names, benchmarks, and research communities. Our synthesis shows recurring fragmentation in terminology, benchmark conventions, and modelling assumptions across the surveyed literatures, but we treat that fragmentation as a qualitative finding of the review rather than as a formal bibliometric result. We also find that CText2SQL currently has the most standardized benchmark and modelling pipeline, whereas manipulation and agentic systems more closely reflect real user workflows but remain harder to evaluate rigorously. Beyond category-specific findings, we identify three cross-cutting themes shared across the field: intent disambiguation and clarification, dialogue context tracking, and evaluation. These reveal a central mismatch between current benchmarks and realistic use: most systems are still optimized for short, clean, single-table interactions rather than long-horizon, ambiguous, multi-source analytical workflows. We conclude by synthesizing the field's main open problems, including unified evaluation, long-dialogue robustness, proactive clarification, interpretability, privacy, domain adaptation, and multi-table reasoning, and argue that progress will depend on moving from narrow task benchmarks toward integrated, user-centered conversational data systems.

1 Introduction

Tabular data, meaning information organized into rows and columns, is one of the most pervasive formats for storing and communicating structured knowledge. Relational databases (Codd, 1970) hold the operational records of governments, hospitals, and enterprises. Spreadsheets have been a primary analytical tool for millions of knowledge workers for decades (Scaffidi et al., 2005). Scientific papers report experimental measurements in tables. Financial disclosures present earnings, revenues, and risk factors in hybrid documents that interleave prose with numerical tables. Enterprise structured data, including relational databases and spreadsheets, accounts for a substantial share of organisational information infrastructure, yet natural language processing (NLP), the scientific study of how computers understand and generate human language, has historically treated *text* as the primary medium of knowledge.

This gap is narrowing. Over the past decade, a growing body of research has asked a deceptively simple question: can a user interact with a table using plain language? Early work framed this as a single-turn *lookup*, where given a question and a table, the task was to find the answer (Pasupat & Liang, 2015). Later work extended this to *translation*, which involves converting a natural language question into SQL (Structured Query Language, a standard language for retrieving data from relational databases) so that answers could be retrieved programmatically (Yu et al., 2018). Both paradigms have produced rapid benchmark progress, with modern systems posting strong results on established evaluations (Gao et al., 2024; Pourreza & Rafiei, 2023).

Yet a fundamental mismatch remains between how these systems are built and how people actually work with data. Real interaction with tabular data is rarely a single question followed by a final answer. A financial analyst studying quarterly earnings does not ask one question and walk away; they probe, follow up, compare across rows, ask for clarification when a figure seems anomalous, request a reformulation when the first answer is ambiguous, and gradually build an understanding of the data through a sequence of exchanges. A physician reviewing a patient's lab results over time asks questions whose meaning depends entirely on what was asked before. A student exploring a dataset for the first time may not even know what question to ask until they have seen a few answers. This kind of iterative, goal directed, multiturn interaction is fundamentally conversational in nature, and it is precisely what existing systems are not designed to support.

1.1 The Conversational Gap

We use the term *conversational* throughout this survey to refer to interactions in which the meaning of a user's current input depends on the history of prior exchanges, and in which the system may itself take communicative actions such as asking for clarification, proposing reformulations, or signaling uncertainty, rather than simply returning an answer. This definition encompasses a range of behaviors: following references to previously mentioned values (such as "how about for the previous quarter?"), interpreting elliptical questions (such as "and the profit margin?" without restating the entity or time period), revising an earlier answer in light of new information, and proactively asking a question when the user's intent is genuinely ambiguous.

The conversational dimension introduces challenges that single-turn systems are not equipped to handle. The system must maintain a *dialogue context*, which is a running representation of what has been said, what has been retrieved or computed, and what the user appears to be trying to accomplish. It must resolve *coreferences*, which are expressions in the current turn that refer to entities, values, or table cells mentioned in a previous turn. It must decide, in real time, whether the user's current question is sufficiently clear to answer or whether asking a clarification question would serve the user better. It must accumulate *numerical context* across turns, since a conversational financial analysis may require an intermediate result computed in turn three to answer the question posed in turn seven. None of these are problems that a single-turn question answering (QA) or Text-to-SQL system faces.

Compounding these challenges, real tables are not clean. They may span multiple pages, contain merged cells and nested headers, mix numerical and categorical entries, embed footnotes, or appear alongside prose that qualifies their contents. A system capable of conversational interaction with such data must handle both the structural complexity of the table and the contextual complexity of the ongoing dialogue simultaneously.

1.2 Related but Fragmented Research Lineages

Research on these problems has not been idle.

Over the past decade, at least five partially overlapping research lineages have been developing systems that address pieces of the conversational table AI problem, though none has framed the full problem explicitly in those terms.

The *conversational table question answering* community has produced datasets such as SQA (Iyyer et al., 2017), which decomposes complex questions about Wikipedia tables into sequences of simpler follow-up questions; HybriDialogue (Nakamura et al., 2022), which grounds multiturn conversations in both tables and the accompanying text passages; PACIFIC (Deng et al., 2022), which introduces proactive clarification

question generation as a component of financial conversational QA; and ConvFinQA (Chen et al., 2022), which requires chained numerical reasoning across the turns of a financial dialogue.

The *conversational Text-to-SQL* community has produced SParC (Yu et al., 2019b) and CoSQL (Yu et al., 2019a), which decompose complex database queries into multiturn dialogues between users and SQL-generating systems, and a rich line of models for tracking SQL context across turns (Ko et al., 2020; Zheng et al., 2022; Zhong et al., 2022).

The *interactive spreadsheet systems* community, operating largely within human-computer interaction (HCI), the study of how people interact with computational systems, has built tools such as SheetCopilot (Li et al., 2023a) and SheetAgent (Chen et al., 2025b) that allow users to modify spreadsheets through natural language instructions executed as sequences of software actions.

The *natural language interfaces to databases* (NLIDBs) community has produced interactive visual analytics tools such as Eviza (Setlur et al., 2016) and NL4DV (Narechania et al., 2021) that allow users to explore data visually through conversational language queries.

Most recently, the *agentic systems* community has developed systems built around large language models (LLMs), which are neural networks trained on large amounts of text that can generate, reason over, and act on language, that autonomously plan and execute sequences of database operations, spreadsheet edits, and data analysis steps in response to open-ended user goals (Zhang et al., 2023b; Tian et al., 2026).

In the taxonomy used in this survey, these lineages condense into four primary task communities plus one enabling layer: the spreadsheet-manipulation and NLIDB/visual-analytics lines are grouped together under *Interactive Table Manipulation*, while *Foundations* is treated separately as the methodological substrate beneath all four. Exploration is therefore tracked as a recurrent interaction pattern within manipulation-oriented systems rather than as a standalone category.

Across the retained corpus, these lineages often frame closely related interaction problems with different terminology, benchmark conventions, and task boundaries. We therefore use *parallel but disconnected* as a careful qualitative characterization of the literature map assembled in this survey: the phrase captures recurring fragmentation in problem framing, benchmark design, and evaluation practice, not a standalone bibliometric claim. A dedicated citation-network study over the surveyed corpus would be a valuable follow-up project, but the present survey does not claim to have established that fragmentation quantitatively.

The distinctive contribution of this survey is therefore not a blanket claim to be the first survey on tables, NL interfaces, or Text-to-SQL in general. It is narrower and more specific: we make multiturn, context-dependent interaction the organizing lens, and we use that lens to read across CTabQA, CText2SQL, interactive manipulation, and agentic table systems while treating Foundations as the shared substrate beneath them.

1.3 What This Survey Contributes

We make the following contributions.

A unified taxonomy. We organize the literature into five categories: (1) *Foundations*, covering methods for representing and encoding tabular data; (2) *Conversational Table QA* (CTabQA), covering multiturn question answering grounded in tables or mixed table-text documents; (3) *Conversational Text-to-SQL* (CText2SQL), covering context-dependent database query generation across dialogue turns; (4) *Interactive Table Manipulation*, covering systems that allow users to modify, create, and explore tables through dialogue; and (5) *Agentic Table Systems*, covering autonomous LLM-based agents that plan and execute multistep operations over structured data.

A comprehensive literature map. We survey 106 unique cited works spanning these five categories, tracing the evolution of each from early dataset construction papers through the latest LLM-based systems. We draw explicit conceptual connections across communities that have often been discussed separately and that still lack shared benchmark, terminology, and evaluation standards.

A structured benchmark analysis. We compare datasets across the five categories along a common set of dimensions, including task type, number of dialogue turns, domain, data size, table source, and evaluation metric, in Table 10. This analysis makes visible both the coverage and the gaps in existing evaluation infrastructure, and reveals that no existing benchmark spans all five categories.

A cross-cutting analysis of shared challenges. We identify three challenges that arise in every category and examine how each community has addressed them: (i) intent disambiguation and clarification, meaning the problem of determining what the user wants when their request is ambiguous; (ii) dialogue context tracking, meaning the problem of maintaining and updating a representation of what has been discussed across turns; and (iii) evaluation, meaning the challenge of measuring progress when task types, data formats, and interaction modes differ substantially across sub-areas.

An agenda for future research. We close with six concrete open problems, grounded in specific gaps identified throughout the survey, that we believe are the highest-leverage targets for the next generation of work in this area.

1.4 Review Protocol and Evidence Policy

This survey is a structured literature review rather than a statistical meta-analysis. In revising the manuscript, we aligned the reporting with the transparency goals of PRISMA 2020 where they fit a computer-science survey that mixes peer-reviewed papers, arXiv preprints, benchmark releases, and system reports (Page et al., 2021).

Search space. We searched ACL Anthology, arXiv, Google Scholar, Semantic Scholar, OpenReview, and benchmark or project pages when a dataset, system release, or public evaluation artifact was part of the evidence chain discussed in the paper. Query families combined terms such as "conversational table question answering," "conversational text-to-sql," "spreadsheet agent," "natural language interface to database," "text-to-visualization," "table agent," and "table foundation model," together with the names of canonical datasets and systems already known to anchor each category.

Time window and corpus freeze. The corpus for this revision was frozen on March 27, 2026. Works appearing after that date are outside scope for the present survey revision.

Inclusion and exclusion. A paper was retained if it made a direct contribution to multiturn or interactive work over structured data, or if it provided background that was necessary to explain one of the five taxonomy categories. We excluded knowledge-graph systems, purely unstructured document dialogue without table grounding, and single-turn table tasks except where they were needed as historical background or as benchmark predecessors.

Preprints and public artifacts. We included preprints when they introduced benchmarks, systems, or public results that had not yet appeared in archival form, which is common in fast-moving parts of the area such as LLM agents and benchmark releases.

Category assignment and retained corpus size. Papers were assigned by primary contribution rather than by every capability they touched. Borderline systems were placed according to their dominant interaction mode and then cross-referenced narratively when they mattered in more than one category. The present section files cite 106 unique works. We verified that count directly from the citation keys used in the manuscript. After the initial search, we also used limited backward snowballing from category-defining datasets, benchmark papers, and survey references when a source was clearly central to a category but not surfaced cleanly by keyword search alone. Borderline papers were adjudicated by asking which interaction mode carried the paper's main experimental burden.

1.5 Scope and Exclusions

The survey's scope is bounded by two criteria. First, the data must be *structured*, meaning organized into rows and columns, whether in a relational database, a spreadsheet, a financial table embedded in a PDF document, or a result table in a scientific paper. We do not cover knowledge graphs, which are graph-structured databases that represent entities and relationships as nodes and edges, as they pose related but distinct challenges and have their own extensive survey literature (Ji et al., 2022). Second, the interaction must be *multiturn or interactive*, involving at minimum the possibility of context-dependent interpretation, coreference resolution, or system-initiated communication beyond a single question and answer exchange. We treat single-turn table QA (Pasupat & Liang, 2015; Nan et al., 2022) and single-turn Text-to-SQL (Yu et al., 2018) as background and motivation rather than primary subject matter.

We exclude table *generation* tasks, where systems synthesize table content from scratch (Fang et al., 2024), and table-based *fact verification* (Chen et al., 2020a), both of which are single-turn in nature and covered by prior surveys. Text-only conversational QA systems such as CoQA (Reddy et al., 2019) and QuAC (Choi et al., 2018) are discussed as structural analogues that helped establish the problem of multiturn QA, but they do not involve tabular grounding and are therefore not primary subject matter.

1.6 Relationship to Prior Surveys

Several recent surveys address adjacent territory, but they organize the area around different primary questions. Fang et al. (2024) review LLMs on tabular data broadly across prediction, generation, question answering, and table understanding. Liu et al. (2024) survey NL-to-SQL methods from a lifecycle perspective centered on translation, data, evaluation, and error analysis. Zhang et al. (2024) survey natural language interfaces for tabular querying and visualization. Katsogiannis-Meimarakis & Koutrika (2023) provide a detailed technical treatment of deep learning Text-to-SQL systems. Zaib et al. (2021) survey conversational QA broadly as a dialogue problem.

Table 1 makes the positioning more explicit. Relative to these surveys, the distinctive value of the present paper is not that it covers every table-related task. It is that it centers multiturn, context-dependent interaction over structured data and then uses that interaction lens to connect querying, SQL generation, manipulation, and orchestration.

This framing leads to a more modest but better-supported claim. Rather than asserting a generic "first to unify," we argue that the paper's contribution is an interaction-centered synthesis across adjacent literatures that are usually surveyed separately.

1.7 Organization

The remainder of this survey is organized as follows. Section 2 establishes background and definitions, including what we mean by a table, what we mean by conversational, and the range of data formats and interaction modalities in scope. Section 3 presents the five-category taxonomy and situates the five research communities within it. Sections 4 through 7 cover each of the five categories in depth. Section 8 examines the three cross-cutting challenges: clarification, dialogue context tracking, and evaluation. Section 9 presents the final research agenda distilled from those cross-cutting themes. Section 10 concludes.

2 Background and Definitions

This section establishes the vocabulary and conceptual boundaries for the rest of the survey. We define what we mean by a table, what forms tabular data takes in practice, what we mean by conversational interaction, and how these two concepts combine to define the scope of this work. Readers already familiar with table-grounded question answering may wish to skim Sections 2.1 and 2.2 and focus on Section 2.3, which introduces the interaction typology that underlies the taxonomy in Section 3.

Table 1: Positioning this survey relative to adjacent surveys. "Multiturn central" asks whether context-dependent dialogue is a primary organizing principle rather than a peripheral topic. "Manipulation / agentic" asks whether spreadsheet actions, tool use, or orchestration over structured data are part of the survey's core scope.

Survey	Primary scope	Multiturn central	Manipulation / agentic	What it does not center
Fang et al. (Fang et al., 2024)	LLMs on tabular data broadly, including prediction, synthesis, question answering, and table understanding	No	No	Multiturn interaction over structured data as the main organizing axis
Liu et al. (Liu et al., 2024)	NL2SQL / Text-to-SQL methods, data, evaluation, and error analysis	Partial	No	Cross-community treatment of QA, manipulation, and agents
Zhang et al. (Zhang et al., 2024)	Natural language interfaces for tabular querying and visualization	Partial	Partial	A taxonomy centered on dialogue state, clarification, and cross-category interaction
Katsogiannis and Koutrika (Katsogiannis-Meimarakis & Koutrika, 2023)	Deep learning Text-to-SQL systems and benchmarks	No	No	Conversational interaction beyond SQL generation
Zaib et al. (Zaib et al., 2021)	Conversational QA broadly as a multiturn dialogue problem	Yes	No	Structured-table interaction as the main object of study
This survey	Conversational AI over structured data across five categories	Yes	Yes	–

2.1 What Is a Table?

For the purposes of this survey, a *table* is any data structure that organizes information into rows and columns, where each column represents a named attribute and each row represents a single entity, observation, or record. This definition is intentionally broad. It covers relational database tables, where rows are called tuples and columns are typed attributes governed by a schema (a formal description of a database's structure, including table names, column names, and data types); spreadsheet grids, where cells may contain numerical values, text, or formulas; financial report tables, which appear embedded in PDF documents and often combine numerical values with textual footnotes; and result tables in scientific publications, which report experimental measurements across conditions.

Not all structures that look like tables share the same properties. Table 2 summarizes the main types of tabular data encountered across the five categories of this survey, along with their characteristic structural properties. Three properties are especially consequential for language-based interaction.

Schema regularity refers to whether the column names and data types are fixed and declared in advance. Relational database tables have highly regular schemas; financial report tables often do not, because column headers may be merged, hierarchical, or implied rather than stated explicitly.

Structural complexity refers to whether the table has a flat two dimensional layout or whether it contains nested headers, merged cells, multirow spans, or embedded subtotals. Wikipedia tables and financial tables frequently exhibit complex structure; CSV (comma-separated values) files, a plain-text format where each row is a line and each field is separated by a comma, and relational database tables are typically flat.

Hybrid content refers to whether the table is the sole carrier of information or whether it appears alongside prose text that qualifies, explains, or extends its contents. Financial reports and scientific papers are hybrid documents; the tables in them cannot be interpreted in isolation from the surrounding text. This matters

Table 2: Types of tabular data encountered across the five categories of this survey. ✓ indicates the property is typically present; ✗ indicates it is typically absent.

Table Type	Regular Schema	Structurally Complex	Hybrid Content
Relational database table	✓	✗	✗
Spreadsheet (office)	✗	✓	✗
Wikipedia table	✗	✓	✓
Financial report table	✗	✓	✓
Scientific result table	✗	✓	✓
CSV / flat file	✓	✗	✗

for conversational systems because a user asking what a footnote means may be referring to text beside the table rather than a cell inside it.

These structural differences have direct consequences for how systems must process tables. A system that works well on flat relational tables may fail on financial report tables with merged column headers. A conversational system that tracks dialogue context over a clean database schema faces qualitatively different challenges from one that must track context over a scientific table whose columns contain abbreviated condition names. Throughout the survey we note, for each dataset and system, which table types are covered.

2.2 What Is Conversational Interaction?

We use the term *conversational* to describe interactions that have at least one of the following four properties.

Context dependence. The meaning of the user's current input cannot be fully determined without reference to one or more prior turns. A question such as "and the year before?" is context dependent: it cannot be answered without knowing what quantity was discussed in the preceding turn. Context dependence is the defining property of multiturn interaction and the primary challenge distinguishing conversational table AI from single-turn table AI.

Coreference. The user's input contains an expression, such as a pronoun or a demonstrative phrase, that refers to an entity, value, column, or row introduced in a prior turn. For example, "what about its operating margin?" uses the word "its" to refer to a company named earlier in the conversation. Resolving such expressions requires the system to maintain a running representation of the entities that have appeared in the dialogue.

System-initiated communication. The system takes a communicative action beyond answering a question. This includes asking a clarification question when the user's intent is ambiguous, proposing an alternative interpretation of a question, or proactively offering information that the user did not ask for but that the system infers would be useful. Systems with this capability engage in what is called *mixed initiative* dialogue, meaning that either participant, user or system, may take the initiative to steer the conversation (Allen et al., 1999).

Iterative refinement. The user progressively adjusts their request across turns, often in response to earlier answers, working toward a single evolving goal rather than issuing independent queries. For example, a user might ask for total revenue in 2023, then request a breakdown by product line, then ask to exclude a particular category from the total. Each turn modifies the scope of the previous request rather than starting fresh.

Not every system surveyed here exhibits all four properties. Earlier dataset-oriented papers primarily address context dependence and coreference. More recent agentic systems engage in iterative refinement and mixed initiative behavior. We use this four-property framework to characterize each system and dataset consistently throughout the survey.

2.3 Interaction Types: A Typology

The taxonomy introduced in Section 3 is organised around four primary modes of interaction between a user and a table-grounded AI system, together with one additional pattern that cuts across them: exploration. Distinguishing these modes before the taxonomy section is important because the differences are not merely technical: they reflect different user goals, different system capabilities, and different notions of what constitutes a correct response.

Querying. The user asks a question and the system returns an answer extracted or computed from the table. The answer may be a single value, a set of values, or a natural language sentence. The user's goal is information retrieval, that is, to learn something from the table that was not already known. Conversational Table QA systems (Section 4) are primarily querying systems.

Translating. The user expresses an information need in natural language and the system translates it into a formal query language, most commonly SQL, which is then executed against a database to produce results. The key difference from querying is that the system's primary output is a program rather than a direct answer. Conversational Text-to-SQL systems (Section 5) are translating systems. The conversational challenge is generating a query whose meaning depends on the history of prior queries in the session.

Manipulating. The user instructs the system to change the contents or structure of the table, for example by inserting rows, deleting entries, reformatting columns, applying formulas, or reorganizing the layout. The user's goal is table modification rather than information retrieval, and a correct response is a modified table that faithfully reflects the user's instruction. Interactive table manipulation systems (Section 6) are manipulation systems.

Exploration as a recurrent interaction pattern. The user does not have a single sharply specified question but wishes to understand the structure, patterns, or anomalies in the data. The system may proactively suggest summaries, highlight unexpected values, propose visualizations, or generate descriptive narratives. This behaviour is the most open ended and the least standardised within the NLP literature. It is most visible in interactive visualization tools such as Eviza (Setlur et al., 2016) and NL4DV (Narechania et al., 2021). In the taxonomy of Section 3, we therefore treat exploration not as a standalone primary category but as a recurrent interaction pattern that most often appears within manipulation-oriented systems and, in practice, can surface in other categories as well.

Orchestrating. The user specifies a high-level goal and the system decomposes it into a sequence of subtasks, executes them autonomously using available tools, and returns a consolidated result. The user's input may be a single sentence that expands into dozens of discrete operations involving multiple tables, code execution, and intermediate reasoning. Agentic table systems (Section 7) are orchestrating systems.

These interaction modes are not mutually exclusive in practice. A real session might begin with exploration, move to targeted querying, include a table edit, and conclude with a compiled report. The taxonomy separates the four primary modes because different research communities have focused on different system outputs, and understanding which mode a given system addresses is essential for interpreting its benchmark results and identifying what it cannot do.

2.4 What Makes Tabular Data Different from Text?

Much of the machinery developed for conversational text understanding, including reading comprehension models, dialogue state trackers, and coreference resolution systems, does not transfer directly to tables. The reasons are worth stating explicitly.

Position encodes meaning. In a table, the meaning of a cell depends on its row and column position. The value 142.3 in a table means nothing without the column header (say, "Revenue in millions") and the row identifier (say, "Q3 2023") that give it meaning. Language models pretrained on running text are not natively aware of this positional structure. Encoding it requires either special pretraining objectives, as in

TaPas (Herzig et al., 2020) and TaBERT (Yin et al., 2020), or deliberate serialization strategies that flatten the table into a text string while preserving positional relationships. Serialization refers to the process of converting a structured data object, such as a table, into a linear sequence of tokens that a language model can process.

Numerical reasoning is often required. Many questions about tables require arithmetic: differences, ratios, percentages, and cumulative sums. Language models are not calculators, and early systems that attempted to answer numerical questions by pattern matching over serialized table text performed poorly on arithmetic-heavy benchmarks such as FinQA (Chen et al., 2021b) and ConvFinQA (Chen et al., 2022). More recent approaches offload arithmetic to a deterministic interpreter by having the model generate executable code or SQL rather than producing a number directly.

Scale varies enormously. A Wikipedia table may contain a few dozen cells. An enterprise database may contain thousands of tables, each with hundreds of columns and millions of rows. A spreadsheet used in financial modeling may span hundreds of sheets with cells containing cross sheet formula references. Techniques that work for small, isolated Wikipedia tables do not necessarily scale to enterprise databases or multisheet spreadsheets.

Tables occur in context. As noted in Section 2.1, many real-world tables are embedded in documents containing prose that explains or qualifies the table's contents. A conversational system over a financial report must understand not just the table but also the surrounding management commentary and the footnotes that redefine certain line items. Strong performance on isolated table benchmarks therefore does not predict strong performance on hybrid document benchmarks, a distinction we return to when comparing datasets in Section 8.

2.5 A Note on Terminology

The literature surveyed here uses inconsistent terminology for similar concepts. We adopt the following conventions throughout this survey.

We use *turn* to mean a single exchange in a conversation: one user utterance and one system response. We use *dialogue* and *conversation* interchangeably for a sequence of turns. We use *session* when we wish to emphasize the full interactive episode, including any accumulated system state.

We use *utterance* for a single natural language input from the user, and *query* when that utterance is directed at a table with the intent to retrieve or compute information. We distinguish *query* in this sense, meaning a natural language expression of an information need, from *SQL query*, meaning a formal expression in Structured Query Language.

We use *grounding* for the process of connecting a natural language expression to specific elements of a table, namely identifying which cells, rows, columns, or values are relevant to a given utterance. Grounding is a prerequisite for answering questions, generating SQL, and producing table edits.

We use *context window* for the portion of the dialogue history that a system actively uses when processing the current turn. In early systems this was limited to the immediately preceding turn or a fixed window of recent turns; in LLM-based systems it may span the entire session subject to the model's maximum input length.

3 A Taxonomy of Conversational AI over Tabular Data

The literature on conversational interaction with tabular and structured data has grown across at least five research communities that have often been studied in partial isolation from one another. Each community developed its own terminology, benchmarks, and evaluation criteria for what is, at a deeper level, a shared problem: enabling users to interact with structured data through natural language over multiple exchanges. Unifying these communities requires a taxonomy that is principled enough to reveal their connections yet specific enough to respect the genuine differences in what each community has built.

We organize the literature into five categories. The first category, *Foundations*, covers methods for representing tables in a form that language models can process; it is the substrate on which the other four categories build. The remaining four categories correspond to the four primary modes of interaction introduced in Section 2.3: querying (CTabQA), translating (CText2SQL), manipulating (Interactive Table Manipulation), and orchestrating (Agentic Table Systems). Exploration is retained in the background typology as a recurrent interaction pattern, but not elevated here to a fifth primary task community. The reason is empirical rather than terminological: exploratory systems such as Eviza and NL4DV are most naturally studied as part of manipulation-oriented and visual-analytics workflows, not as a mature standalone research community with its own benchmarks and modelling lineage. We therefore track exploration as a design dimension that can appear within manipulation systems and, in practice, across other categories as well.

Three cross-cutting themes arise in every category and are treated separately in Section 8: intent disambiguation and clarification, dialogue context tracking, and evaluation. Figure 1 provides a visual overview of the taxonomy. Table 3 gives a compact characterization of each category.

3.1 A Running Example Across the Taxonomy

Consider a user working with a quarterly sales workbook who asks the following composite request: "Which product line had the sharpest margin decline after Q2, flag every quarter below 15% margin in the spreadsheet, and prepare a short note for the CFO explaining the result."

The *Foundations* layer determines how the system represents the workbook in the first place: headers, rows, formulas, and nearby text must be encoded so that the model can recover the table's structure and semantics faithfully.

CTabQA covers the question-answering part of the request. The system must identify the relevant rows, columns, and calculations needed to answer which product line declined most, possibly over multiple follow-up turns such as "only for Europe" or "show me the underlying numbers."

CText2SQL covers the same intent when the underlying data lives in a relational database rather than directly in the visible spreadsheet. In that setting, the follow-up turns must be translated into context-dependent SQL that preserves earlier constraints and updates them correctly.

Interactive Table Manipulation covers the state-changing part of the request. Here the system is no longer only answering a question; it must edit the artifact itself by inserting a flag column, writing formulas, or generating a derived table or visualization.

Agentic Table Systems cover the end-to-end orchestration problem. An agent may need to decide whether clarification is required, retrieve the correct data source, execute analysis steps, apply the spreadsheet edits, and draft the CFO note while keeping provenance visible.

The example is intentionally simple, but it shows why the five categories are complementary rather than competing. They describe different layers and interaction modes within a single broader problem: helping users work with structured data through dialogue.

3.2 Category 1: Foundations

The Foundations category covers methods that enable language models to process tabular data at all. These are not conversational systems in themselves; rather, they provide the representational substrate on which conversational systems are built. We include them as a distinct category because advances in table representation have consistently unlocked new capabilities in the categories above them.

The central challenge addressed by this category is that language models are trained on sequences of tokens, meaning words, subwords, and characters arranged in a linear order. Tables are not linear. A cell's meaning depends on its column and row position simultaneously, and a model that reads a serialized table as if it were running text may fail to recover this two-dimensional structure. Two broad lines of work have addressed this problem.

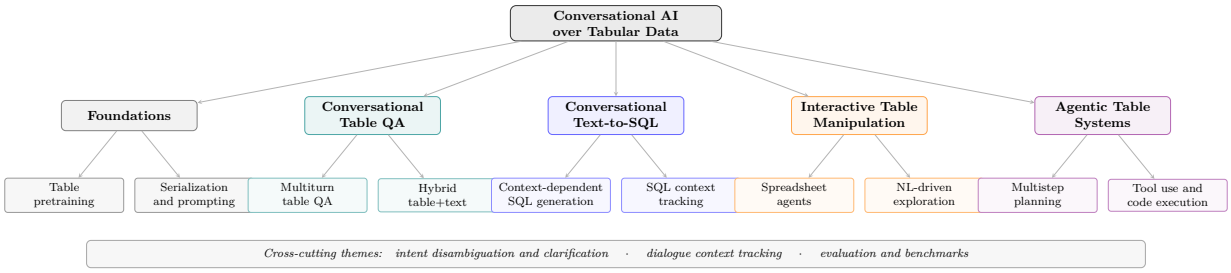

Figure 1: Taxonomy of conversational AI over tabular and structured data. Five categories are arranged below the root node, each with two representative sub-areas. CTabQA = Conversational Table Question Answering; CText2SQL = Conversational Text-to-SQL. The italicised bar at the bottom represents three cross-cutting themes that arise in every category, analysed together in Section 8.

Table 3: Summary of the five taxonomy categories. The interaction mode column uses the typology defined in Section 2.3. Representative papers are listed to orient the reader before the detailed sections that follow.

Category	Mode	Primary Task	Representative Works	Section
Foundations	enabling substrate†	Table encoding, serialization, pretraining	TaPas (Herzig et al., 2020); TaBERT (Yin et al., 2020); TAPEX (Liu et al., 2022b); TableGPT2 (Su et al., 2024)	3.2
Conversational Table QA (CTabQA)	Querying	Multiturn natural language question answering over tables or hybrid documents	SQA (Iyyer et al., 2017); HybriDialogue (Nakamura et al., 2022); PACIFIC (Deng et al., 2022); ConvFinQA (Chen et al., 2022)	4
Conversational Text-to-SQL (CText2SQL)	Translating	Context-dependent natural language to SQL generation across dialogue turns	SParC (Yu et al., 2019b); CoSQL (Yu et al., 2019a); RASAT (Zhong et al., 2022); CoE-SQL (Wang et al., 2024a)	5
Interactive Table Manipulation	Manipulating (+ exploration)	Natural language driven modification, exploration, and generation of table content	SheetCopilot (Li et al., 2023a); SheetAgent (Chen et al., 2025b); Eviza (Setlur et al., 2016)	6
Agentic Table Systems	Orchestrating	Multistep autonomous planning and execution over structured data	Data-Copilot (Zhang et al., 2023b); InfiAgent-DABench (Hu et al., 2024); AutoTQA (Zhu et al., 2024)	7

† Foundations is not itself one of the four primary interaction modes defined in Section 2.3; it is included separately because representation, serialization, and table pretraining methods enable all four modes.

Table pretraining adapts language model pretraining to make models sensitive to tabular structure. TaPas (Herzig et al., 2020) extends BERT (Bidirectional Encoder Representations from Transformers, a widely used pretrained language model (Devlin et al., 2019)) with additional token-type (segment) embeddings that encode each token's row index and column index within the table, allowing it to answer table questions through weak supervision, meaning training using automatically derived rather than manually annotated labels. TaBERT (Yin et al., 2020) introduces vertical self-attention, a mechanism that connects tokens occupying the same column position across different rows. TAPEX (Liu et al., 2022b) takes a distinct approach, pretraining a model to function as a neural SQL executor by training it to produce the output of SQL queries over tables. OmniTab (Jiang et al., 2022) combines natural table–text pairs with synthetic question answering supervision, while UnifiedSKG (Xie et al., 2022) broadens the picture by framing 21 structured knowledge grounding tasks in a unified text-to-text setting rather than as isolated problems. More recent large-scale models such as TableGPT2 (Su et al., 2024) train dedicated table encoders jointly

Table 4: Representative methods in the Foundations category. "Training style" is used instead of "pretraining objective" because some methods are primarily inference-time frameworks rather than newly pretrained models. Input/access format: Flat = sequence serialization; Struct = structure-aware encoding; RAG = retrieval-augmented; ICL = in-context learning.

Method	Year	Training style	Input/access	Scale	Key contribution
TAPAS (Herzig et al., 2020)	2020	MLM + weak supervision	Flat + table embeddings	BERT-Large	Extends BERT to table–text inputs with row and column embeddings for cell selection and aggregation
TaBERT (Yin et al., 2020)	2020	MLM + MCP + CVR	Struct	BERT	Joint text–table pretraining with vertical attention over rows and column-aware encoding
TAPEX (Liu et al., 2022b)	2022	SQL-exec pretraining	Flat	BART-Large	Trains a seq2seq model to mimic SQL execution for table reasoning
OmniTab (Jiang et al., 2022)	2022	Natural + synthetic pretraining	Flat	BART-Large	Combines web table–text data and SQL-derived synthetic QA in one table QA framework
UnifiedSKG (Xie et al., 2022)	2022	Unified text-to-text multitask learning	Struct	T5-3B	Frames 21 structured knowledge grounding tasks in a single text-to-text setup
Binder (Cheng et al., 2023)	2023	Training-free neural-symbolic ICL	Code + table	Codex	Generates executable programs with API calls to support symbolic reasoning over tables
Chain-of-Table (Wang et al., 2024b)	2024	ICL with table operations	Struct + op history	LLM-based	Uses intermediate table transformations as the reasoning chain instead of free-form CoT
TableRAG (Chen et al., 2024)	2024	RAG + program-aided reasoning	RAG	LLM-based	Scales table QA with query expansion plus schema and cell retrieval
TableGPT2 (Su et al., 2024)	2024	Encoder alignment + instruct tuning	Struct + encoder	7B / 72B	Couples a semantic table encoder with an LLM for stronger table-centric reasoning

with a large language model, combining structural table understanding with the broad language capabilities of modern LLMs.

Serialization and prompting addresses how to present a table to a language model at inference time, which is the stage where a trained model processes new inputs rather than updating its weights. Approaches range from simple linearization, in which row contents are concatenated as a text string, to structured prompting strategies such as Chain-of-Table (Wang et al., 2024b), which interleaves table operations with the model's intermediate reasoning steps, and Binder (Cheng et al., 2023), which connects the model to a symbolic execution environment so that arithmetic is handled by a code interpreter rather than the model itself. TableRAG (Chen et al., 2024) introduces a retrieval-augmented approach that allows models to selectively access relevant portions of very large tables rather than encoding the entire table in a single input, addressing the scale challenge discussed in Section 2.4. Table 4 provides a structured comparison of the key Foundations methods covered in this category.

The specific methods from this category are discussed inline as they arise in Sections 4 through 7, where each is introduced in the context of the conversational system that builds on it. The enabling-substrate status of Foundations is defined once in Table 3 and is not repeated in later tables.

3.3 Category 2: Conversational Table QA

Conversational Table QA (CTabQA) is the core category of this survey. It covers systems designed to answer natural language questions where the questions form a multiturn dialogue and are grounded in one or more tables, or in hybrid documents that combine tables with prose text.

What distinguishes CTabQA from single-turn table QA is that each question in the dialogue may be incomprehensible without the history of prior questions and answers. Questions may refer back to previously mentioned values, ask for comparative or follow-up analysis of something computed in an earlier turn, or be deliberately elliptical, omitting the subject or object that the user has in mind, which must be recovered from context.

CTabQA has developed primarily through dataset construction. The Sequential QA dataset (SQA, Iyyer et al., 2017) established the task by decomposing complex Wikipedia-table questions into sequences of simpler sub-questions. Later datasets introduced richer dialogue phenomena: HybriDialogue (Nakamura et al., 2022) grounds conversations in both tables and the text passages that accompany them; PACIFIC (Deng et al., 2022) adds proactive clarification, where the system must decide when a question is ambiguous enough to warrant asking the user for clarification rather than guessing; ConvFinQA (Chen et al., 2022) requires multistep numerical reasoning across turns in financial documents; and iTBLS (Sundar et al., 2025) introduces interpretation, modification, and generation as three distinct conversational tasks over scientific tables, making it a bridge between QA-oriented and manipulation-oriented settings.

CTabQA is reviewed in detail in Section 4.

3.4 Category 3: Conversational Text-to-SQL

Conversational Text-to-SQL (CText2SQL) covers systems that translate natural language questions into SQL queries, where successive questions in a dialogue may modify, refine, or extend the SQL generated in prior turns. This is arguably the most technically mature of the five categories, with well-established benchmarks and a clear progression of model architectures from rule-based predecessors to the current generation of LLM-based systems.

The defining challenge is *SQL context*: the query that correctly answers the current question may differ only slightly from the query that answered the previous question, and the user often relies on the system to carry that context forward without being told to do so explicitly. A user who asked for total revenue in 2022 and then asks "and the breakdown by region?" is implicitly requesting a variant of the previous query, not an independent one from scratch. Systems that treat each question independently will generate incorrect SQL for such dependent questions.

The field is anchored by two benchmark datasets: SParC (Yu et al., 2019b), which provides expert-annotated question sequences over 200 databases derived from the single-turn Spider benchmark (Yu et al., 2018), and CoSQL (Yu et al., 2019a), which uses a Wizard-of-Oz collection protocol. In this protocol, one human plays the role of a database system and another plays the role of a user; the recorded interaction produces more naturalistic conversations that include clarification exchanges alongside SQL-bearing turns.

Model approaches have evolved from SQL-editing methods (Zhang et al., 2019; Ko et al., 2020) that represent how the current SQL differs from the previous one, through relation-aware encoder architectures (Zhong et al., 2022) that inject coreference and schema relations into the model's attention mechanism (the component of a neural network that determines which parts of the input to weight most when producing an output), to LLM-based prompting strategies (Wang et al., 2024a; 2025) that leverage in-context learning, which is the ability of large language models to perform a task by reading worked examples in the input without updating model weights.

CText2SQL is reviewed in detail in Section 5.

3.5 Category 4: Interactive Table Manipulation

Interactive Table Manipulation covers systems where the user's goal is not to retrieve information from a table but to modify, restructure, or extend the table itself through natural language instructions. This category is the closest to what most people do with spreadsheets in practice: not querying them but editing them.

The primary application domain is office productivity. Systems such as SheetCopilot (Li et al., 2023a) and SheetAgent (Chen et al., 2025b) allow users to instruct a spreadsheet agent to insert rows, apply conditional formatting, build pivot tables (summary tables that reorganize and aggregate data from a source table by grouping rows and computing aggregate values such as sums or averages), generate charts, or write cell formulas, all through natural language. A key technical challenge is translating natural language instructions into sequences of atomic software actions, which are indivisible single operations such as setting a cell value, sorting a column, or deleting a row, that when executed in order produce the desired table state.

This category also includes interactive visualization and data exploration systems such as Eviza (Setlur et al., 2016), as well as toolkits such as NL4DV (Narechania et al., 2021) that map natural language utterances to analytic specifications. In these settings, the user explores data visually and the system responds by updating a chart or generating a new visualization in response to each natural language utterance. The conversational aspect here is that successive visualizations are contextually linked: asking "zoom in on the outliers" makes sense only with reference to the chart currently on screen.

Interactive Table Manipulation is reviewed in detail in Section 6.

3.6 Category 5: Agentic Table Systems

Agentic Table Systems covers the most open-ended category: systems where a user provides a high-level goal and an LLM-based agent plans and executes a sequence of operations autonomously, using tools such as SQL execution environments, Python code interpreters, spreadsheet APIs (application programming interfaces, which are defined communication channels that allow software systems to interact with each other), and web search as needed.

The key distinction from the other four categories is the degree of *system autonomy*. In the previous four categories, the user explicitly controls the granularity of each step by asking a question, issuing a SQL request, or giving a specific manipulation instruction. In agentic systems, the user expresses a goal and the system works out what steps are needed. This places agentic systems at the intersection of conversational AI, program synthesis (the automated generation of programs that satisfy a given specification or goal), and tool-augmented reasoning.

Representative systems include Data-Copilot (Zhang et al., 2023b), which builds an autonomous workflow for heterogeneous data analysis; InfiAgent-DABench (Hu et al., 2024), which benchmarks LLM agents on realistic data analysis tasks over tabular data; and AutoTQA (Zhu et al., 2024), which uses multiagent collaboration to answer complex tabular questions. The D-Bot system (Zhou et al., 2024) applies agentic reasoning to database diagnosis, demonstrating that the agentic paradigm extends beyond data retrieval to database operations and maintenance.

A recent survey by Tian et al. (2026) characterizes LLM-based table agents along five competency dimensions: structure understanding, semantic understanding, retrieval and compression, executable reasoning, and cross-domain generalization. Our survey complements this engineering-focused work by situating agentic systems within the broader conversational context and examining how they relate to the other four interaction modes.

Agentic Table Systems are reviewed in detail in Section 7.

3.7 Connections Across Categories

The five categories are not isolated silos. Several connections run between them, and understanding these connections reveals both the current structure of the field and its most productive future directions.

Foundations enables everything. Every system in categories 2 through 5 relies, explicitly or implicitly, on the table representation and serialization techniques developed in category 1. The choice of how to encode a table, whether through a specialised pretraining objective, a structured serialization format, or a retrieval-based compression scheme, affects performance across all downstream tasks. Improvements in Foundations therefore have broad leverage.

CTabQA and CText2SQL serve the same user need differently. Both categories address users who want to retrieve information from a table using natural language. The difference is in what the system produces: a natural language answer (CTabQA) or an executable SQL query (CText2SQL). In practice, these approaches are increasingly hybridized. Some recent systems for table QA and numerical reasoning generate intermediate SQL or Python code as a reasoning step before producing a natural language response (Cheng et al., 2023; Chen et al., 2022), blurring the categorical boundary.

Interactive Manipulation adds a write dimension. Categories 2 and 3 are read-only: the user queries a table that does not change. Category 4 introduces writing, where the table itself is modified. This changes the nature of dialogue context significantly, because the system must track not just what was asked and answered but what the table currently looks like, a state that depends on all prior manipulation actions in the session.

Agentic Systems subsume and extend. An agentic system handling an open-ended data analysis goal may internally invoke querying, SQL generation, and table manipulation as sub-tasks within a single session. This makes agentic systems among the most powerful and the hardest to evaluate because, to our knowledge, no existing benchmark evaluates all of these capabilities within a single unified conversational framework.

All categories share three cross-cutting challenges. Regardless of whether the task is answering a question, generating SQL, editing a spreadsheet, or orchestrating a multistep analysis, every category must address: how to resolve ambiguous user intent, how to maintain and update dialogue context across turns, and how to measure progress when task definitions, data formats, and interaction modes vary across sub-areas. These three challenges are the subject of Section 8.

4 Conversational Table Question Answering

Conversational Table Question Answering (CTabQA) is the most direct expression of the survey's central problem: answering table-grounded questions whose interpretation depends on dialogue history. It covers systems that answer natural language questions grounded in tables or in hybrid documents that combine tables with prose text, where the questions form a dialogue in which the meaning of each question may depend on what was asked and answered before. This dependence is what separates CTabQA from single-turn table QA: a user does not issue one question and receive one final answer, but engages in an exchange that progressively builds an understanding of the data.

The research challenges in CTabQA are qualitatively different from those in single-turn table QA. A system must track which entities, values, and columns have been mentioned across turns. It must interpret questions that omit information recoverable only from prior context. It must carry numerical results forward across turns rather than computing each answer independently. And in the most ambitious systems, it must decide when the user's intent is ambiguous enough to warrant asking a clarification question rather than guessing.

We organise this section as follows. Section 4.1 situates CTabQA relative to its text-only predecessors. Section 4.2 surveys the benchmark datasets, which have been the primary driver of progress in this sub-area. Section 4.3 enumerates the core technical challenges. Section 4.4 surveys the main modelling approaches. Section 4.5 discusses domain-specific directions. Section 4.6 identifies the current limits of the field.

4.1 Background: From Text-Only Conversational QA to Tables

CTabQA did not emerge in isolation. It inherited both its problem formulation and its evaluation conventions from a well-developed tradition of conversational QA over free text.

Two datasets established the conventions of conversational QA over text and are widely used as baselines and design references. CoQA (Reddy et al., 2019) contains approximately 8,000 multiturn dialogues grounded in passages from seven domains, from children's literature to Reddit posts, and requires the model to produce free-text answers that depend on conversation history. QuAC (Choi et al., 2018) frames conversational QA as an information-seeking dialogue in which a student asks questions about a Wikipedia passage without seeing it, while a teacher with full access provides answers; this setting makes the asymmetry between the user's partial knowledge and the system's full access to the data explicit. Both datasets established that multiturn QA is a qualitatively harder task than single-turn QA, and that models trained on single-turn benchmarks do not generalise well to conversational ones.

The transition to tabular grounding introduced three additional complications. First, the structure of a table is not linear: a model that treats a serialized table as running text loses the positional meaning encoded in rows and columns. Second, many questions about tables require arithmetic, and text-only conversational QA models are not equipped for numerical reasoning. Third, hybrid documents, in which tables are embedded in paragraphs of explanatory prose, require the model to reason jointly over structured and unstructured data simultaneously.

Single-turn table QA datasets such as WikiTableQuestions (Pasupat & Liang, 2015), which contains compositional questions over Wikipedia tables, and FeTaQA (Nan et al., 2022), which requires generating free-form answers from tables, established that structured data poses challenges beyond what text QA models handle. HybridQA (Chen et al., 2020b) and OTT-QA (Chen et al., 2021a) pushed further by requiring multihop reasoning that traverses both a table and linked Wikipedia passages before arriving at an answer, foreshadowing the hybrid grounding demands that became central in CTabQA. CTabQA builds on all of this infrastructure while adding the conversational dimension.

4.2 Datasets

Progress in CTabQA has been driven primarily by dataset construction. Each new dataset introduced one or more phenomena, such as clarification, numerical chaining, or hybrid grounding, that prior datasets did not cover. Table 5 provides a structured comparison of all CTabQA datasets discussed in this section.

The datasets are evaluated with a mix of answer-level metrics (e.g., exact match and token-level F1), numerical reasoning metrics (execution accuracy and program accuracy), ranking metrics for retrieval (e.g., MRR, MAP, NDCG), and text-generation metrics (e.g., ROUGE, BLEU, METEOR, and BERTScore). Here, execution accuracy measures whether the predicted program yields the correct answer when run, whereas program accuracy measures whether the generated program itself matches the reference program.

Sequential QA (SQA). The Sequential QA dataset (Iyyer et al., 2017) was the first to frame multiturn table QA as a standalone task. It contains 6,066 question sequences (17,553 question-answer pairs) decomposed from complex single questions about Wikipedia tables: each sequence breaks a compositional question into two or three simpler sub-questions that must be answered in order. For example, a single question such as "which team won the most games in the decade with the highest overall attendance?" becomes a sequence: first, identify the decade with the highest attendance; then, identify the winning team within that decade. SQA established the decomposition paradigm and is typically evaluated with overall accuracy as well as sequence accuracy, the stricter metric that requires every turn in a sequence to be answered correctly. Its limitation is that the question sequences are scripted by annotators who already knew the answer, making them less naturalistic than dialogues collected from real users.

HybriDialogue. HybriDialogue (Nakamura et al., 2022) contains 4,844 multiturn conversations grounded in Wikipedia pages that contain both tables and the prose text surrounding them. The dialogues were crowdsourced by decomposing the multihop questions in OTT-QA (Chen et al., 2021a) into sequences of simpler questions that each require evidence from either the table, the text, or both. This hybrid evidence structure makes HybriDialogue the most direct predecessor to real-world conversational data analysis, where a user may ask about a table value in one turn and follow up with a question whose answer appears in a nearby paragraph. The dataset evaluates three sub-tasks: retrieval (identifying which table and passage are relevant to each turn), state tracking (maintaining a running list of retrieved evidence), and response

Table 5: Comparison of CTabQA-related datasets. Size is reported using each dataset's native counting unit(s). Evidence: T = table, T+X = table plus text, M = multimodal (e.g., text + tables + images, or paper components such as tables, figures, and equations). Eval. lists the primary evaluation families used in the original paper. † TAT-QA and FinQA are single-turn predecessors included because PACIFIC and ConvFinQA are derived from them, respectively.

Dataset	Year	Size	Domain	Evidence	Eval.
SQA (Iyyer et al., 2017)	2017	6,066 seq. (17,553 QAs)	Wikipedia	T	Overall acc., seq. acc.
HybriDialogue (Nakamura et al., 2022)	2022	4,844 conv.	Wikipedia	T+X	MRR/MAP; Sacre-BLEU/BERTScore
PACIFIC (Deng et al., 2022)	2022	2,757 conv. (19,008 turns)	Finance	T+X	CNP: P/R/F1; CQG: ROUGE-2/EM/F1; CQA/PCQA: EM/F1
ConvFinQA (Chen et al., 2022)	2022	3,892 conv. (14,115 Qs)	Finance	T+X	ExeAcc, ProgAcc
MMCoQA (Li et al., 2022)	2022	1,179 conv. (5,753 QAs)	General	M	Recall/NDCG; F1/EM
cPAPERS (Sundar et al., 2024)	2024	5,030 QA pairs	Scientific	M	ROUGE, METEOR, BERTScore, BLEU
iTBLS (Sundar et al., 2025)	2025	4,000 tables	Scientific	T	EM
TAT-QA (Zhu et al., 2021)†	2021	16,552 Qs (2,757 ctx)	Finance	T+X	EM, F1
FinQA (Chen et al., 2021b)†	2021	8,281 examples	Finance	T+X	ExeAcc, ProgAcc

generation (producing a natural language answer), with ranking metrics for retrieval/state tracking and generation metrics such as SacreBLEU and BERTScore for response quality.

PACIFIC. PACIFIC (Deng et al., 2022) introduces a qualitatively new phenomenon: *proactive clarification*. The dataset contains 2,757 dialogues with 19,008 question-answer turns grounded in hybrid financial contexts derived from TAT-QA (Zhu et al., 2021), where each example combines numerical tables with supporting prose text. In addition to answering questions, the system must decide when a question is sufficiently ambiguous to warrant generating a clarification question (CQG) rather than attempting an answer. PACIFIC also requires numerical reasoning, since financial questions frequently ask for percentage changes, ratios, or multistep arithmetic across table values. The task is evaluated across its sub-components: clarification need prediction with precision/recall/F1, clarification question generation with ROUGE-2, exact match, and token-level F1, and conversational answering with exact match and numeracy-focused F1. PACIFIC is the most directly relevant CTabQA dataset to the concerns of this survey because it is the only dataset that makes the clarification decision itself a first-class modelling objective.

ConvFinQA. ConvFinQA (Chen et al., 2022) extends the single-turn FinQA benchmark (Chen et al., 2021b) to the conversational setting. It contains 3,892 multiturn conversations over financial reports from the S&P 500 (the Standard and Poor's 500, a widely used index of 500 large companies listed on American stock exchanges), totalling 14,115 question-answer pairs. Each conversation chains a sequence of numerical reasoning steps across turns: the answer computed in one turn often becomes an implicit operand (a value used in a calculation) in a later turn. For example, a user might first ask for operating income, then ask for

the ratio of operating income to total revenue, using the previously computed value without restating it. This *numerical chaining* across turns is the defining challenge of ConvFinQA and makes it substantially harder than single-turn numerical reasoning benchmarks. ConvFinQA is typically evaluated with both execution accuracy and program accuracy, reflecting whether the predicted reasoning program yields the correct result and whether the program itself matches the reference.

MMCoQA. MMCoQA (Li et al., 2022) extends conversational QA to multimodal evidence sources: each dialogue is grounded in a combination of text passages, tables, and images, and the model must identify which modality contains the answer at each turn. The multimodal evidence setting reflects the reality of many documents, such as product manuals or annual reports, where tables, figures, and text are all used simultaneously to convey information. MMCoQA challenges models to perform both cross-modal retrieval (finding the relevant evidence across modalities) and cross-turn reasoning (interpreting the current question in light of the conversation history).

iTBLS. The iTBLS dataset (Sundar et al., 2025) is the most recent and the most directly aligned with the interactive scope of this survey. It contains conversations over tables extracted from arXiv preprints, covering three distinct task types within the same conversational framework: *interpretation* (asking the system to explain or summarise table contents), *modification* (asking the system to describe how the table should be changed), and *generation* (asking the system to produce new table content consistent with a given context). This three-task structure acknowledges that real users do not interact with tables purely through questions but also through instructions and requests for explanation. The QA-reformulation approach introduced in the same work converts modification and generation requests into question-answer pairs to leverage existing QA models. The paper reports up to 13% improvement in exact match on iTBLS itself and up to 16% improvement in BERTScore when applying the reformulation approach to prior text-to-table benchmarks. BERTScore (Zhang et al., 2020) measures semantic similarity between the system's output and a reference answer using contextual token embeddings from a pretrained language model, rather than requiring exact string match.

cPAPERS. cPAPERS (Sundar et al., 2024) collects conversational question-answer pairs from scientific paper reviews and grounds them in paper components such as tables, figures, and equations. It comprises 5,030 question-answer pairs, including 1,601 table-grounded pairs, and is evaluated with text-generation metrics such as ROUGE, METEOR, BERTScore, and BLEU. As a companion dataset to iTBLS, it extends the scientific-domain focus to the review process itself, reflecting the practical use case of a researcher asking a review assistant to help evaluate claims made in a paper.

Single-turn predecessors: TAT-QA and FinQA. TAT-QA (Zhu et al., 2021) contains 16,552 question-answer pairs over hybrid financial documents and introduced the challenge of reasoning jointly over tabular numbers and textual qualifications in the same document. FinQA (Chen et al., 2021b) established numerical reasoning over financial reports as a benchmark task, requiring models to produce a sequence of arithmetic operations that derive the answer rather than extracting it directly. Neither is multiturn, but both are cited throughout the CTabQA literature as the single-turn foundations on which ConvFinQA and PACIFIC were built, with ConvFinQA derived from FinQA and PACIFIC extending the hybrid financial setting introduced by TAT-QA.

4.3 Core Technical Challenges

Across all CTabQA datasets, four technical challenges recur with sufficient frequency to warrant explicit discussion. Understanding where each current system succeeds and fails along these dimensions is more informative than a bare performance comparison.

Context-dependent question interpretation. A question such as "and what about the previous year?" is fully interpretable only by a model that has retained what quantity was being discussed and what year was the most recent reference. This requires the system to maintain a representation of the *dialogue context*: the entities, values, columns, and time references introduced across prior turns. Systems that compress dialogue

context by concatenating prior question-answer pairs as a text prefix perform reasonably on short dialogues but degrade on long ones, because the model must attend over increasingly large inputs to find the relevant context (Chen et al., 2022).

Coreference resolution over tables. Pronouns and demonstrative expressions that refer to table cells or column headers across turns are a persistent challenge. In text-only conversational QA, coreference resolution benefits from decades of research on entity tracking in prose (Clark & Manning, 2016). In tabular settings, the referent is often a structured entity defined by its row and column coordinates rather than a named entity that appears as a text string, making standard coreference approaches difficult to apply directly.

Multiturn numerical reasoning. The numerical chaining pattern central to ConvFinQA and PACIFIC requires the system to use intermediate results from prior turns as implicit inputs to later computations, without those intermediate values being restated. This is qualitatively different from single-turn numerical reasoning, where all operands are present in the input. Current LLM-based approaches address this by generating executable code that explicitly stores intermediate values as named variables, allowing later turns to reference them by name rather than by recomputing from the raw table (Chen et al., 2022).

Proactive clarification. Deciding whether to answer a question or ask for clarification requires the system to assess its own uncertainty about the user's intent. This is an inherently probabilistic judgment: the system must estimate whether its most likely interpretation of the question is likely enough to be correct, or whether the cost of answering incorrectly exceeds the cost of asking for clarification. PACIFIC is the only CTabQA dataset that makes this decision explicit, and existing PACIFIC results show that proactive clarification remains a challenging open problem (Deng et al., 2022).

4.4 Modelling Approaches

CTabQA models have evolved from early pipeline architectures toward end-to-end systems that leverage large language models. The evolution reflects both the maturation of the underlying language model technology and the increasing complexity of the datasets that defined what a solution must look like.

Retrieval-then-generate pipelines. HybriDialogue introduced a retrieval-then-generate paradigm for conversational Tabular QA: the system first retrieves the relevant table (and, in hybrid settings, a supporting passage), then generates an answer conditioned on the retrieved evidence and the conversation history (Nakamura et al., 2022). The cTBLS system (Sundar & Heck, 2023) instantiates this paradigm with a three-stage architecture. First, a dense table retriever encodes the current user turn (with dialogue context) and candidate tables into a shared vector space and selects the most relevant table by vector similarity. Second, a coarse-to-fine cell ranking (system-state tracking) module narrows attention to the specific rows and columns relevant to the current question. Third, a large language model (GPT-3.5) generates the final response conditioned on the ranked cells and the dialogue history. On HybriDialogue table retrieval, cTBLS reports a Top-1 accuracy improvement from 0.345 (BM25) to 0.777 (cTBLS-DTR), i.e., a $((0.777 - 0.345)/0.345) \approx 125\%$ relative gain, alongside improvements in MRR@10 ($0.491 \rightarrow 0.846$) and Top-3 accuracy ($0.460 \rightarrow 0.901$) (Table 1 in Sundar & Heck, 2023). For response generation, the best cTBLS configuration reported achieves ROUGE-1/2/L *precision* of 0.642/0.322/0.548, compared to 0.438/0.212/0.375 for the HybriDialogue baseline reported in the same evaluation table (Table 4 in Sundar & Heck, 2023). ROUGE (Recall-Oriented Understudy for Gisting Evaluation) measures lexical overlap between n-grams (contiguous sequences of n words) in the generated response and a reference answer.

Sequence-to-sequence approaches with multitask learning. UniPCQA, introduced as the baseline model in PACIFIC (Deng et al., 2022), is a sequence-to-sequence model (a model that reads a sequence of tokens and produces a sequence of tokens as output) trained jointly on the answer generation task and the clarification question generation task. Multitask learning here refers to training a single model to perform multiple distinct tasks simultaneously, with the expectation that shared representations across tasks will improve performance on each individual task compared to training separate models. UniPCQA also reformulates numerical reasoning as code generation: rather than producing a numerical answer directly,

the model produces a small program that, when executed, yields the answer, offloading arithmetic to a deterministic interpreter.

Sequence tagging for numerical extraction. TAGOP, introduced as the baseline in TAT-QA (Zhu et al., 2021), frames QA over hybrid financial documents as a sequence tagging problem: the model tags each token in the concatenated table-text input as either relevant or irrelevant to the answer, and then applies an operator (sum, difference, percentage change, and so on) to the tagged values. This approach is effective for single-turn numerical QA but does not extend naturally to multiturn settings where the relevant values change across turns.

LLM-based end-to-end systems. The availability of large language models capable of in-context learning has enabled a shift from task-specific architectures to general-purpose systems that are instructed to perform CTabQA through a combination of system prompt, serialized table, and conversation history. EHRAgent (Shi et al., 2024) demonstrates this paradigm in the medical domain. It frames conversational QA over Electronic Health Records (structured clinical databases containing patient measurements, diagnoses, and treatments) as a code generation task: the model generates Python code that queries the underlying database, executes it, and returns the result. This approach separates language understanding from computation and avoids the hallucination risk (the tendency of language models to generate plausible-sounding but incorrect information) that arises when a model attempts to perform arithmetic directly over serialized table text.

Recent prompting-based LLM work also provides a useful bridge from the earlier task-specific literature to the evaluation discussion in Section 8.3. Across established CTabQA benchmarks such as SQA and ConvFinQA, later systems increasingly rely on prompt engineering, retrieval, or program execution rather than fully task-specific architectures (Sui et al., 2024b; Khatuya et al., 2025). This shift helps explain why the latest competitive systems are often LLM-mediated even though much of the category's core dataset and model design work predates the current LLM wave.

4.5 Domain-Specific Directions

CTabQA research has concentrated heavily in two application domains: finance and science. This concentration reflects both the availability of high-quality annotated data in these domains and the genuine need for AI assistance with complex tables in professional settings.

Finance. The financial domain has produced the densest cluster of CTabQA datasets: TAT-QA (Zhu et al., 2021), FinQA (Chen et al., 2021b), ConvFinQA (Chen et al., 2022), PACIFIC (Deng et al., 2022), and MultiHiertt (Zhao et al., 2022), which requires numerical reasoning over tables that are themselves hierarchically organised with subtotals and category headers. The common thread is that financial documents combine numerical tables with prose commentary in a way that requires joint reasoning, and that the questions asked by professional analysts frequently require multistep arithmetic and references to prior context. The financial domain is also a high-stakes setting: an incorrect answer about an operating margin or a debt ratio can have material consequences, making reliability and interpretability of answers especially important.

Science. The iTBLS (Sundar et al., 2025) and cPAPERS (Sundar et al., 2024) datasets focus on scientific tables, specifically experimental result tables from arXiv preprints. Scientific tables differ from financial tables in their column structure: they report experimental conditions, model names, and evaluation metric values, often with numerous abbreviations that require domain knowledge to interpret. The practical use case targeted by these datasets is a researcher asking an AI assistant to help analyse or compare results across papers.

Medicine. EHRAgent (Shi et al., 2024) demonstrates conversational table QA in the medical domain, where the tables are Electronic Health Record (EHR) databases containing patient lab results, vital signs, diagnoses, and medication records. Broader medical QA benchmarks such as MedQA (Jin et al., 2020) motivate the need for reliable clinical question answering, though most existing medical QA benchmarks are

single-turn and are not grounded in structured EHR tables. The domain presents unique challenges related to privacy, the need for precise numerical reasoning over clinical measurements, and the high cost of errors.

4.6 Category-Specific Remaining Gaps

The unresolved issues in CTabQA are now fairly specific to the task formulation itself. Most benchmarks still assume a single table or single document per dialogue, so the category remains weak on multi-table, multi-document, and cross-period analysis. The strongest datasets also concentrate on answer correctness, leaving cell grounding, supporting evidence selection, and stepwise numerical faithfulness under-measured. Clarification appears explicitly only in a small part of the literature, most notably PACIFIC, rather than as a standard expectation for table-grounded QA. These are the local bottlenecks for CTabQA; Section 8 synthesises the shared patterns that recur across categories, and Section 9 turns that synthesis into the final agenda.

5 Conversational Text-to-SQL

Conversational Text-to-SQL (CText2SQL) is the most technically mature category in this survey. It addresses the task of translating a user's natural language question into a SQL query (a formal expression in Structured Query Language used to retrieve data from a relational database), where the correct translation of the current question may depend on the SQL queries generated in prior turns of the same dialogue.

The field is defined by a clear benchmark progression, a well-studied set of technical challenges, and an architecture trajectory that moves from hand-crafted SQL-editing rules through graph-based neural encoders to the current generation of large language model prompting approaches. This maturity makes CText2SQL the best-understood sub-area in conversational table AI, but it also means that its benchmarks and evaluation conventions have calcified in ways that limit generalization to real-world settings.

We organise this section as follows. Section 5.1 traces the single-turn foundations. Section 5.2 describes the benchmark datasets. Section 5.3 enumerates the core technical challenges. Section 5.4 surveys the main modelling approaches organized by architectural generation. Section 5.5 covers data augmentation and cross-lingual extensions. Section 5.6 identifies current limits and open directions.

5.1 Background: From Single-Turn to Multiturn Text-to-SQL

Text-to-SQL as a research task predates the modern deep learning era. The ATIS line of evaluation, including ATIS-3 (Dahl et al., 1994), established the classic single-domain setting around air-travel queries. In its later semantic parsing and Text-to-SQL use, ATIS became the canonical benchmark for mapping natural language requests such as "show me all flights from Boston to Atlanta on Tuesday" into formal database queries. ATIS demonstrated that natural language database interfaces were feasible in restricted domains, but its single-domain nature meant that models trained on it did not generalize to new databases.

The introduction of Spider (Yu et al., 2018) in 2018 shifted the field decisively toward cross-domain Text-to-SQL. Spider contains 10,181 questions over 200 databases spanning 138 domains, with a training and test split that ensures no database appears in both sets. This cross-domain generalization requirement, the ability to generate correct SQL for a database the model has never seen during training, became the defining challenge of the modern era. Single-turn models trained on Spider improved rapidly: RAT-SQL (Wang et al., 2020) introduced relation-aware schema encoding and linking; PICARD (Scholak et al., 2021) added constrained decoding (a technique that restricts the model's output at each step to tokens that form syntactically valid SQL, preventing the generation of malformed queries); and LGESQL (Cao et al., 2021) used line graphs (graph structures that represent relationships between edges of another graph) to model both local and non-local schema relations. By 2024, few-shot prompting approaches such as DAIL-SQL (Gao et al., 2024) and decomposed in-context learning approaches such as DIN-SQL (Pourreza & Rafiei, 2023) had pushed execution accuracy on Spider to 86.6% (DAIL-SQL), within approximately six percentage points of the estimated human performance ceiling of 92.6%, a substantial advance, though meaningful headroom to human-level performance remains.

Table 6: Comparison of datasets relevant to CText2SQL, ordered by year. Context indicates whether the correct output depends on prior turns or iterative workflow context. EX denotes execution accuracy. EM denotes Spider-style exact set match. QM and IM denote question-level and interaction-level exact set match, respectively. SR denotes success rate. DA denotes dialogue-act prediction accuracy. LCR denotes logic correctness rate.

Dataset	Year	Size	Context	DBs	Collection	Primary Metrics
ATIS (Dahl et al., 1994)	1994	8,297 train utt.; 3,211 test utt.	Partial	1	Multi-site spoken dialogue	Task-specific ATIS evaluation
Spider (Yu et al., 2018)	2018	10,181 Q; 5,693 SQL	No	200	Student annotators	EM, EX
SParC (Yu et al., 2019b)	2019	4,298 interactions; 12k+ Q	Yes	200	Controlled user interactions	QM, IM
CoSQL (Yu et al., 2019a)	2019	3,007 dialogues; 30k+ turns; 10k+ SQL	Yes	200	Wizard-of-Oz	QM, IM, BLEU, LCR, DA acc.
BIRD (Li et al., 2023b)	2023	12,751 Q–SQL pairs	No	95	Expert-curated	EX
Spider 2.0 (Lei et al., 2025)	2024	632 workflow problems	Multi-step	213	Expert-curated enterprise workflows	SR (Spider 2.0); EX (lite/snow)

The maturation of single-turn Text-to-SQL raised a natural question: real database users do not issue isolated questions. They engage in sessions where each question refines or extends the previous one. The CText2SQL task was introduced to address exactly this gap, and its defining datasets, SParC and CoSQL, were built directly on the Spider database collection to enable direct comparison with single-turn baselines.

Comprehensive surveys of single-turn Text-to-SQL approaches include Katsogiannis-Meimarakis & Koutrika (2023), which covers deep learning methods through 2023, and the natural language interface survey by Liu et al. (2024). The present section focuses exclusively on the multiturn conversational extension.

5.2 Datasets

Two benchmark datasets define the core CText2SQL landscape: SParC and CoSQL. Both are built on the Spider database collection, which enables direct comparison with single-turn Text-to-SQL baselines and isolates the additional difficulty introduced by conversational context. For reference, Table 6 also includes historically important or practically relevant non-conversational Text-to-SQL benchmarks discussed in this section.

SParC. SParC (Semantic Parsing in Context, Yu et al., 2019b) is the primary CText2SQL benchmark. It contains 4,298 interaction sequences derived from Spider questions, with 12,166 individual questions spread across 200 databases. Expert annotators constructed the sequences by designing question chains that progressively explore a database topic: a sequence might begin with a broad question about all employees, then narrow to a department, then filter by salary, then ask for a count. Each question in a sequence was annotated with its gold SQL, making SParC suitable for turn-level evaluation.

SParC is evaluated using two metrics. Question match (QM) measures Spider-style exact set match at the turn level: a turn is counted as correct only if the predicted SQL matches the gold SQL under the benchmark's clause-level exact-match evaluator, not by raw string identity. Interaction match (IM) is stricter: a full interaction is counted as correct only if every turn in it achieves QM. IM directly measures end-to-end conversational correctness but is so strict that most systems score well below 50% on it even when turn-level accuracy is substantially higher.

CoSQL. CoSQL (Yu et al., 2019a) was collected using the Wizard-of-Oz protocol applied to the Spider databases. Pairs of crowd workers were recruited: one played the role of a user exploring a database, and

the other acted as a SQL expert with full access to the database schema, answering questions by writing and executing SQL queries. The user could not see the database schema (the formal description of table names and column names) and had to ask questions in natural language; the system worker provided answers and, when necessary, asked clarification questions. This asymmetric setup produces dialogues that are more naturalistic than SParC's expert-designed sequences because the user's questions emerge from genuine information-seeking rather than from a pre-planned decomposition.

CoSQL adds two tasks beyond SQL state tracking: response generation from database results and user dialogue act prediction. This makes CoSQL historically important within CText2SQL because it evaluates not only whether the system recovers the correct SQL, but also whether it can communicate appropriately during database interaction.

BIRD and Spider 2.0. BIRD (Li et al., 2023b) is a single-turn benchmark but is widely used in the CText2SQL literature as a measure of difficulty relative to Spider. Its 12,751 questions are more complex than Spider's, requiring external knowledge (facts not present in the database schema) and involving larger, dirtier databases. BIRD's difficulty ceiling serves as a reference for evaluating how close CText2SQL systems are to handling realistic enterprise queries.

Spider 2.0 (Lei et al., 2025) takes this realism further by using actual enterprise database environments involving multi-database workflows, nested queries, and operations that go beyond standard SELECT statements. Neither BIRD nor Spider 2.0 is multiturn, but both signal that the gap between current CText2SQL benchmarks and real-world database interaction remains large.

5.3 Core Technical Challenges

The CText2SQL task inherits all the challenges of single-turn Text-to-SQL and adds several that are specific to the conversational setting.

Schema linking across turns. Schema linking is the task of identifying which tables and columns in the database schema are relevant to a given natural language question. In single-turn Text-to-SQL, schema linking is applied once per question. In CText2SQL, the set of relevant schema elements can change across turns as the user shifts focus. A question such as "now look at the managers instead" implies a change in the relevant table without naming it. The system must track which schema elements have been in scope and which are being introduced or abandoned at each turn.

SQL context tracking. The SQL generated for the current question often differs from the SQL generated for the previous question by only one or two clauses. A question that adds a filter condition (a WHERE clause, which specifies that only rows satisfying a condition should be returned) to the previous query, or that changes an aggregation function (a function such as COUNT, SUM, or AVG that computes a single summary value from a set of rows), requires the system to identify the prior SQL and modify it correctly. Systems that generate SQL from scratch for each question, ignoring the prior SQL, will fail to produce the correct incremental modification.

Implicit references and ellipsis. Users frequently issue questions that are grammatically incomplete because they rely on the conversational context to fill in the missing parts. A question such as "what about in 2022?" cannot be answered without knowing what attribute was being discussed, what aggregation was being applied, and what database table was in scope. Resolving these implicit references, a phenomenon called ellipsis (the omission of words recoverable from context), requires the system to maintain a representation of the current conversational state.

Clarification and dialogue act recognition. CoSQL demonstrates that real users sometimes issue questions that are ambiguous given the database schema, questions for which more than one SQL query could be a reasonable interpretation. A system that always guesses will generate incorrect SQL for ambiguous questions. A system that asks for clarification unnecessarily will frustrate users. Deciding when to ask for

Table 7: Representative CText2SQL systems organized by architectural generation, reported with a *consistent exact-match metric family*. QM = question match and IM = interaction match. Because the cited papers do not all report on the same split, we explicitly list the evaluation split used in each source. For Track-SQL, results depend on the backbone; we report the stronger DeepSeek-7B variant reported in the cited paper. CoE-SQL reports official test-set execution-based question and interaction scores separately in the original paper, so we discuss those figures in prose below instead of mixing them directly into this exact-match table. MAC-SQL is included for completeness but does not report SParC/CoSQL results in the cited paper.

System	Generation	Core Mechanism	SParC QM	SParC IM	CoSQL QM	CoSQL IM	Split
Edit-SQL (Zhang et al., 2019)	SQL editing	Copies prior SQL and edits it for the current turn	47.9	25.3	40.8	13.7	test
IGSQL (Ko et al., 2020)	Graph encoder	Schema interaction graph across turns	51.2	29.5	42.5	15.0	test
IST-SQL (Wang et al., 2021)	State tracking	Explicit schema-state and SQL-state tracking	47.6	29.9	41.8	15.2	SParC dev, CoSQL test
HIE-SQL (Zheng et al., 2022)	History-enhanced encoder	History-aware bimodal encoder with schema linking	64.6	42.9	53.9	24.6	test
RASAT (Zhong et al., 2022)	Relation-aware	T5 with relation-aware self-attention and coreference relations	67.7	49.1	58.8	27.0	test
ACT-SQL (Zhang et al., 2023a)	LLM prompting	In-context learning with automatically generated chain-of-thought	51.0	24.4	46.0	13.3	dev
CoE-SQL (Wang et al., 2024a)	LLM prompting	Chain-of-editions prompting from prior SQL	56.0	36.5	52.4	23.9	dev
Track-SQL (Chen et al., 2025a)	Dual-extractive	Separate schema extractor and context extractor modules	65.17	46.44	58.19	28.67	dev
MAC-SQL (Wang et al., 2025)	Multi-agent	Decomposition, tool use, and refinement by collaborating agents	N/A	N/A	N/A	N/A	N/A

clarification requires the system to model its own uncertainty about the correct interpretation, a capability that most current CText2SQL systems do not have.

Error propagation across turns. In a multiturn session, an incorrect SQL generated for one turn will mislead the system for all subsequent turns that depend on it. If the system generated the wrong column in turn two and the user's turn-three question says "and sort by that column", the system may compound the original error rather than detecting it. Error propagation is particularly damaging in long dialogues and has motivated research into interaction-state tracking methods that maintain an explicit record of which SQL components are well-established and which are uncertain (Wang et al., 2021).

5.4 Modelling Approaches

CText2SQL models have evolved through three identifiable generations. Table 7 summarises the key systems discussed in this section.

Generation 1: SQL-editing models (2019 to 2021). The earliest CText2SQL systems addressed the SQL context challenge by representing the current question as an edit applied to the prior SQL query. Edit-SQL (Zhang et al., 2019) was the first to make this explicit: rather than generating SQL for each turn from scratch, it conditions the decoder on the previously generated SQL and produces a sequence of edit operations (insertions, deletions, and substitutions of SQL tokens) that transform the prior query into the current one. This approach correctly captures the intuition that successive questions in a dialogue often differ by only a small number of SQL modifications. Its limitation is that it relies on the prior SQL being correct; if an earlier turn produced an error, the editing approach will propagate that error forward.

Generation 2: Graph-based encoders (2020 to 2022). The second generation shifted from output-side SQL editing to input-side context modeling. These systems represent the dialogue history and the database schema as a graph and use graph neural networks (neural networks that operate on graph-structured data, propagating information between connected nodes) to jointly encode the schema, the conversation history, and the current question.

IGSQL (Ko et al., 2020) introduced the schema interaction graph, which adds edges between schema nodes across turns to represent which tables and columns have been referenced together in prior queries. HIE-SQL (Zheng et al., 2022) augmented this with a history-enhanced encoder that explicitly models coreference relations between the current question's tokens and entities mentioned in prior turns. IST-SQL (Wang et al., 2021) maintained an explicit interaction state: a running summary of which schema elements are active, which columns have been filtered on, and which aggregations are in scope, updated at each turn.

RASAT (Zhong et al., 2022) is the strongest model of this generation. It builds on T5, a pretrained sequence-to-sequence language model (a model trained on large text corpora that reads a sequence of input tokens and produces a sequence of output tokens), and injects coreference relations as explicit relation types in the self-attention mechanism. The self-attention mechanism assigns weights to pairs of input tokens based on their relevance to each other; RASAT extends it to also consider whether two tokens are related through a coreference link that spans dialogue turns. RASAT achieved state-of-the-art performance on both SParC and CoSQL at the time of publication.

Generation 3: Large language model approaches (2022 to present). The availability of large language models with strong in-context learning capabilities shifted the modelling paradigm from finetuning task-specific architectures to prompting general-purpose models with carefully designed examples.

ACT-SQL (Zhang et al., 2023a) demonstrated that a well-structured prompt, containing the database schema, the dialogue history, a small set of annotated examples, and the current question, is sufficient to elicit competitive SQL generation from a large language model without any task-specific finetuning. Finetuning refers to the process of continuing to train a pretrained model on task-specific data to adapt its weights; prompting bypasses this by providing the task context entirely within the model's input.

CoE-SQL (Wang et al., 2024a) introduced chain-of-editions prompting, which instructs the model to generate each turn's SQL as an explicit edit of the prior turn's SQL, directly encoding the SQL context challenge into the prompt format. In the cited paper, this yields 56.0 QM and 36.5 IM on the SParC development split, and the official test table also reports execution-based question and interaction scores of 74.1 and 51.9 on SParC and 71.1 and 42.9 on CoSQL. Those test-set numbers are not exact-match figures, so they should not be read as directly interchangeable with the QM/IM rows in Table 7; they are best interpreted as complementary evidence that the chain-of-editions prompt transfers well to execution-based evaluation.

Track-SQL (Chen et al., 2025a) takes a different approach: rather than relying on a single model to jointly handle schema linking and context tracking, it uses two separate extraction modules, one for schema elements and one for context-dependent tokens, whose outputs are combined before the SQL generator sees them. This modular design isolates two sources of error, and the cited paper reports 65.17 QM / 46.44 IM on SParC and 58.19 QM / 28.67 IM on CoSQL on the development split with a DeepSeek-7B backbone. These are the cleanest late-generation exact-match-family numbers discussed in this section, although they remain development-set rather than official hidden-test results.

Multi-agent approaches. MAC-SQL (Wang et al., 2025) decomposes the Text-to-SQL task among multiple agents: a selector agent identifies the relevant tables, a decomposer agent breaks a complex question into simpler sub-questions, and a refiner agent checks and corrects the generated SQL. Although originally designed for single-turn Text-to-SQL, the multi-agent architecture is directly relevant to CText2SQL because the decomposer's sub-question strategy mirrors the natural structure of a conversational session. This connection between multi-agent decomposition and multiturn dialogue is an active area of investigation.

CHASE-SQL (Pourreza et al., 2025) and Alpha-SQL (Li et al., 2025) extend multi-agent and search-based reasoning to the SQL generation problem. CHASE-SQL uses multi-path reasoning and preference optimization (a training technique that adjusts model outputs toward preferred completions based on human or automated feedback) to improve SQL quality. Alpha-SQL applies Monte Carlo Tree Search (a planning algorithm that selects actions by simulating many possible futures and choosing the action with the best average outcome) to zero-shot SQL generation, treating SQL construction as a sequential decision problem. Both are primarily evaluated in single-turn settings but represent approaches whose structure extends naturally to conversational SQL generation.

Viewed across these modelling families, CText2SQL also has a useful, but limited, connection to dialogue state tracking (DST). The analogy helps explain the design space without defining a separate modelling lineage: SQL-editing systems treat the next turn as a state update over the previous query; question-rewriting systems externalise context resolution by converting an elliptical turn into a self-contained one; and IST-SQL (Wang et al., 2021) comes closest to explicit state maintenance over a structured interaction history. What is still missing is a widely adopted query-state representation that separates confirmed constraints from inferred ones and supports repair when later turns reveal that an earlier assumption was wrong. We therefore use the DST analogy here as an interpretive lens on the models above, and return to its broader cross-cutting significance in Section 8.2.2.

5.5 Data Augmentation and Cross-Lingual Extensions

A persistent limitation of CText2SQL research is data scarcity. Constructing multiturn Text-to-SQL datasets requires annotators who understand both natural language and SQL, and who can design interaction sequences that test a specific conversational phenomenon. This expertise is expensive, which is why SParC and CoSQL together contain fewer than 8,000 dialogues, a much smaller training set than comparable single-turn benchmarks.

Liu et al. (2022a) introduced self-play data augmentation as a response to this scarcity. A SQL-to-text model (a model that generates natural language questions from SQL queries) and a Text-to-SQL model are alternated: the SQL-to-text model generates new questions for unseen databases, and the Text-to-SQL model generates their SQL annotations. This bootstrapping procedure produces training data for new databases without requiring human annotators, and the models trained on the augmented data generalize better to new database schemas.

Question rewriting (Vakulenko et al., 2021) offers a complementary approach. A context-dependent question such as "and what about the previous year?" is rewritten into a fully self-contained question that incorporates the missing context explicitly, such as "what was the total revenue in 2021?" The rewritten question can then be handled by any single-turn Text-to-SQL system without modification. Question rewriting separates the conversational context resolution problem from the SQL generation problem, making it easier to plug in improved single-turn SQL generators as they appear. Its limitation is that the rewriter itself must handle coreference and ellipsis, shifting the difficulty rather than eliminating it.

Cross-lingual extensions illustrate another open front for CText2SQL. Most existing work focuses on English, but database users worldwide issue queries in many languages. The structural challenges of multiturn SQL generation therefore compound with the linguistic challenges of cross-lingual transfer (applying a model trained primarily on one language to queries in another language).

5.6 Category-Specific Remaining Gaps

CText2SQL is comparatively mature, but its main remaining gaps are tightly tied to semantic parsing over realistic databases. Its benchmarks still underrepresent messy enterprise schemas, incomplete metadata, and deployment settings where schema exposure, privacy, and access control matter. A second local weakness is that clarification is not yet deeply integrated into SQL state tracking: systems can often recover the next query, but they are less reliable at deciding when ambiguity warrants asking the user and then constraining later SQL with the answer. Finally, strong per-turn generation still does not fully solve interaction-level consistency across long query sequences. These are the local bottlenecks for CText2SQL; the broader recurring themes are synthesised once in Section 8 and then condensed in Section 9.

6 Interactive Table Manipulation

The previous two sections covered systems that *read* tables: they answer questions or generate SQL in response to user queries, but the underlying table does not change. Interactive Table Manipulation covers a qualitatively different mode of interaction: the user instructs the system to *change* the table. The output of a successful interaction is not an answer but a modified table, a new visualization, or a generated piece of table content that reflects the user's intent.

This category encompasses two lines of research that developed largely independently. The first, rooted in NLP and human-computer interaction (HCI), addresses spreadsheet agents: systems that accept natural language instructions and translate them into sequences of operations that modify a spreadsheet. The second, rooted in the visualization community, addresses natural language interfaces for data exploration: systems that allow users to iteratively refine data visualizations through natural language. Both lines share the property that interaction is genuinely bidirectional: the user's input changes the artifact (spreadsheet or visualization) that the system displays, and subsequent user inputs are interpreted in the context of that changed artifact.

We organise this section as follows. Section 6.1 situates the category relative to prior work in natural language interfaces and HCI. Section 6.2 characterises the manipulation task space. Section 6.3 describes the evaluation benchmarks. Section 6.4 surveys spreadsheet manipulation systems. Section 6.5 surveys interactive visualization and data exploration systems. Section 6.6 enumerates the shared technical challenges. Section 6.7 identifies current limits and open directions.

6.1 Background: Natural Language Interfaces and HCI

The idea of controlling software through natural language commands predates the modern deep learning era by several decades. Early natural language interfaces to databases, such as LUNAR (Woods, 1973) and INTELLECT, allowed users to query datasets through typed English but required hand-crafted grammar rules that were brittle and domain-specific. The challenge of translating natural language commands into executable software operations has always been harder than translating them into SQL queries, because the action space of a general-purpose application, such as a spreadsheet, is far larger and less formally specified than the SQL grammar.

The maturation of large language models reopened this problem. LLMs trained on code and on API documentation can generate function calls and script sequences for applications such as Microsoft Excel, Google Sheets, and Python data analysis libraries including pandas (a widely used Python library for tabular data manipulation) and matplotlib (a library for generating plots and charts), without requiring hand-crafted grammars. This capability has driven a rapid expansion of research into LLM-based spreadsheet agents since 2023.

On the visualization side, the HCI community established interactive natural language interfaces for data exploration through systems such as Eviza (Setlur et al., 2016) and NL4DV (Narechania et al., 2021) in the 2016 to 2021 period, before the current LLM wave. These systems demonstrated that users could effectively explore datasets through iterated natural language requests, each refining the visualization produced by the previous one. Their evaluation methods, centred on user studies and analytic specification accuracy, differ

substantially from the benchmark-driven evaluation used in the NLP community, a divergence that makes cross-community comparison difficult.

6.2 The Manipulation Task Space

Interactive table manipulation can be organised along two dimensions: the *target* of the operation and the *type* of operation applied.

Targets include cells, rows, columns, worksheets, charts, pivot tables, and, in transformation settings, intermediate or output tables. The same natural language instruction can imply different executable actions depending on the target. For example, "remove duplicates" applied to a column means deduplicating values within that field, whereas the same instruction applied to rows means deleting repeated records.

Operation types include entry (setting or clearing values), management (inserting, deleting, reordering, or creating structural units such as rows, columns, and sheets), formatting (changing appearance or display format), formula writing (inserting spreadsheet formulas), data transformation (reshaping or synthesising tables, including pivoting, unpivoting, melting, or merging), and chart generation (creating or modifying visualisations from tabular data).

Table 8 summarises representative systems in this space. The landscape is still fragmented. General-purpose spreadsheet agents cover parts of entry, management, formatting, charting, or formula-related work, but broad end-to-end support remains limited, especially for long-horizon tasks and transformation-heavy workflows. By contrast, systems such as Rigel and NL2Rigel specialise in table restructuring, while Eviza, NL4DV, and Orko focus primarily on chart specification or interactive visual analysis rather than cell-level spreadsheet editing.

6.3 Benchmarks

Unlike CTabQA and CText2SQL, which converged on a small number of community-standard benchmarks, Interactive Table Manipulation has a fragmented evaluation landscape. Each major system introduced its own benchmark, making direct comparison across systems difficult.

SheetCopilot Benchmark (SCB). The SheetCopilot Benchmark (SCB) contains 221 spreadsheet control tasks distributed across 28 Excel workbooks, derived from spreadsheet-related Q&A (e.g., SuperUser) and adapted to a fixed workbench (Li et al., 2023a). To analyse task diversity, SheetCopilot defines six categories: entry and manipulation, management, formatting, charts, pivot tables, and formulas. Each task specifies a starting spreadsheet state and a natural language instruction; the evaluation executes the generated action sequence and compares the resulting sheet state against one of the ground-truth solutions using an automated comparator.

SCB reports multiple metrics. Exec@1 measures whether the generated action sequence executes without throwing exceptions, while Pass@1 measures functional correctness of the final spreadsheet state. SheetCopilot also reports efficiency metrics (A50 and A90), which summarise the ratio between the number of generated atomic actions and the number in a reference solution.

SheetRM. SheetRM (Chen et al., 2025c) was introduced alongside SheetAgent to better reflect real-life spreadsheet manipulation challenges. Compared with SCB, which primarily targets short-to-medium horizon tasks, SheetRM emphasises long-horizon, multi-category tasks where later steps depend on earlier intermediate results and where choosing the correct operation often requires reasoning over spreadsheet content. In addition to Exec@1 and Pass@1, SheetRM reports SubPass@1, which measures the fraction of subtasks completed correctly within a multi-step task and helps distinguish partial progress from total failure.

SpreadsheetBench. SpreadsheetBench (Ma et al., 2024) is a challenging benchmark introduced at NeurIPS 2024 (Datasets and Benchmarks Track). It contains 912 real instructions gathered from four online Excel forums (ExcelForum, Chandoo, MrExcel, and ExcelGuru), paired with spreadsheets that reflect

Table 8: Representative systems in the Interactive Table Manipulation category and closely related chart-oriented interfaces. Operation types covered: E = entry/value editing, M = structural or worksheet management, F = formatting, W = formula writing, T = table transformation (including pivot-style restructuring), C = chart generation or chart interaction. "Benchmark / Eval." denotes the primary benchmark or evaluation setting reported by the paper. For user-study systems, the final column summarises the main reported quantitative finding rather than Pass@1. N/A indicates that a directly comparable benchmark score was not reported.

System	Year	Operations	Benchmark / Eval.	Backbone	Main Result
SheetCopilot (Li et al., 2023a)	2023	E, M, F, W, T, C	SCB	GPT-3.5 / GPT-4	55.0 Pass@1[†]
SheetAgent (Chen et al., 2025b)	2025	E, M, F, T, C	SCB, SheetRM	GPT-3.5	61.1 Pass@1 (SCB); 44.8 Pass@1 / 77.0 SubPass@1 (SheetRM)
SheetMind (Zhu et al., 2025)	2025	E, M, T	Self-curated tasks	Gemini	80% single-step; ∼70% multi-step success
TableLLM (Zhang et al., 2025)	2025	E, M, T, C	Modified WikiSQL, filtered Spider, self-created op. benchmark	Llama 3.1-8B	89.6 (WikiSQL mod.); 77.8 (op. bench.)
Rigel (Chen et al., 2023)	2023	T	User study	Rule-based	N/A
NL2Rigel (Huang et al., 2024a)	2024	T	Comparative user study	GPT-3.5	Same completion rate as Rigel (86/96), with lower time cost
Eviza (Setlur et al., 2016)	2016	C	User study	Rule-based	N/A
NL4DV (Narechania et al., 2021)	2021	C	Toolkit / case studies	Rule-based	N/A
Orko (Srinivasan & Stasko, 2018)	2018	C	User study	Rule-based	N/A

[†] Reported on a subset of SCB with GPT-4. On the full SCB benchmark, SheetCopilot reports 44.3 Pass@1 with GPT-3.5.

real-world complexity (e.g., multiple tables, non-standard relational layouts, and non-textual elements). To reduce false positives from solutions overfitting a single spreadsheet instance, SpreadsheetBench adopts an online-judge style protocol: each instruction is evaluated on multiple input–output test cases (2,729 total, ∼3 per instruction), and a solution is counted as correct only if it passes all associated cases.

The benchmark exposes a large gap between current agents and human performance. In the authors' evaluation, even strong systems such as Copilot in Excel achieve only around 20% accuracy, and GPT-4o scores around 17% under single-round inference, indicating that realistic formula writing and transformation-heavy tasks remain far from solved.

Visualization benchmarks. The visualization systems in this category are evaluated through user studies and analytic specification accuracy rather than programmatic task completion. NL4DV (Narechania et al., 2021) evaluates accuracy of the analytic specification (a formal description of what visualization to produce, including the data attributes to map to visual channels such as x-axis, y-axis, and color) generated from a natural language query. Eviza (Setlur et al., 2016) and Orko (Srinivasan & Stasko, 2018) were evaluated primarily through user studies measuring task completion time, error rate, and subjective satisfaction. The absence of a shared programmatic evaluation standard for natural language driven visualization is a recurring limitation noted in the visualization literature.

6.4 Spreadsheet Manipulation Systems

SheetCopilot. SheetCopilot (Li et al., 2023a) is a seminal LLM-based spreadsheet manipulation system, published at NeurIPS 2023. It introduces an atomic action library as a virtual API for spreadsheet software and a state machine-based planning framework that iterates in an observe–propose–revise–act loop. At each step, the language model proposes the next action given a textual serialization of the current sheet state; the proposal can be revised using documentation and, when available, execution feedback.

On the SheetCopilot Benchmark (SCB), SheetCopilot attains 44.3% Pass@1 (87.3% Exec@1) on the full benchmark with GPT-3.5. For less accessible models such as GPT-4, results are reported on a 10% subset of SCB, where GPT-4 reaches 55.0% Pass@1. Despite these gains, performance degrades on longer-horizon tasks and on cases that require substantial reasoning about table content or non-standard layouts, motivating subsequent work on more realistic benchmarks and modular agent designs.

SheetAgent. SheetAgent (Chen et al., 2025b) extends SheetCopilot with a three-module architecture. The Planner decomposes the user's instruction into a subtask sequence, the Informer selects and retrieves the spreadsheet context relevant to the current subtask, and the Retriever maps the plan to executable API calls. This separation of concerns reduces context load per step and helps the system scale to larger and more complex spreadsheets.

Using `gpt-3.5-turbo-0613` as the backbone, SheetAgent achieves 61.1% Pass@1 (94.1% Exec@1) on SCB, improving over SheetCopilot's 44.3% Pass@1 on the same benchmark. On the long-horizon SheetRM benchmark, the main reported SheetAgent (GPT-3.5) result is 44.8% Pass@1 with 77.0% SubPass@1 (89.3% Exec@1), highlighting that many subtasks can be completed correctly while end-to-end completion of realistic multi-step workflows remains difficult.

SheetMind. SheetMind (Zhu et al., 2025) is an end-to-end multi-agent framework deployed as a Google Sheets Workspace extension. It comprises three specialised agents: a Manager agent that decomposes complex user instructions into subtasks, an Action agent that translates subtasks into structured commands constrained by a Backus–Naur Form (BNF) grammar, and a Reflection agent that checks alignment between generated actions and the user's original intent. Grammar-constrained generation reduces ill-formed action sequences, while the reflection stage provides a lightweight form of self-checking before execution.

On the benchmarks reported by the paper, SheetMind achieves an 80% success rate on single-step tasks and approximately 70% success on multi-step instructions.

SpreadsheetLLM. SpreadsheetLLM (Tian et al., 2024) targets a practical bottleneck shared by many spreadsheet assistants: real spreadsheets are often too large to fit within a model's context window. Rather than proposing a full manipulation agent, SpreadsheetLLM introduces a structured compression scheme that preserves the spreadsheet's layout and semantics while reducing token length, for example by identifying anchor cells, pruning redundant regions, and encoding type patterns instead of raw values. This kind of representation learning is complementary to manipulation agents: it can be used to provide more faithful context to planning or reasoning modules when operating over large workbooks.

TableLLM. TableLLM (Zhang et al., 2025) takes a fine-tuning approach rather than relying solely on prompting a proprietary model. It fine-tunes Llama 3.1-8B on office-scenario tabular manipulation tasks, covering operations such as querying, updating, merging across tables, and chart generation for tables embedded in documents or spreadsheets. Because the backbone is open and can be deployed locally, TableLLM directly targets privacy and compliance constraints that arise when sensitive office documents are sent to external APIs.

In evaluation, TableLLM reports results on modified WikiSQL (89.6), filtered Spider (81.1), and a self-created operation benchmark (77.8), indicating strong performance on classical text-to-SQL benchmarks alongside a dedicated manipulation setting.

Rigel. Rigel (Chen et al., 2023) addresses data transformation using a declarative mapping approach. Rather than generating imperative step-by-step instructions (commands that specify *how* to perform an operation), Rigel allows users to specify their desired output table structure declaratively (describing *what* the output should look like) using natural language, and the system infers the sequence of reshape operations needed to produce it. This is particularly useful for transformations such as pivoting (rotating a table so that unique values in one column become new column headers), unpivoting (the reverse of pivoting), and joining (combining rows from two tables based on matching values in a shared column).

NL2Rigel. Huang et al. (2024a) present NL2Rigel, an interactive system that synthesises and refines table transformations from natural language instructions. Given structured (or semi-structured) input data, NL2Rigel uses a large language model (reported with GPT-3.5) and prompting to translate user intent into a declarative transformation pipeline expressed in Rigel specifications. The interface visualises the pipeline and intermediate results, enabling users to debug and iteratively refine the transformation with targeted follow-up instructions. A comparative user study reports comparable task completion accuracy to using Rigel directly (86/96 tasks completed in both conditions), while reducing time cost for most tasks.

6.5 Interactive Visualization and Data Exploration

The visualization sub-community developed natural language interfaces for data exploration independently of the NLP spreadsheet agent literature, with a different set of technical approaches and evaluation conventions. These systems share the core conversational property that each user utterance is interpreted relative to the current visualization state, but they focus on the chart rather than the underlying table as the artifact being modified.

Eviza. Eviza (Setlur et al., 2016) is one of the earliest conversational natural language interfaces for visual data analysis, published at UIST 2016. It allows users to issue natural language queries about a scatterplot or bar chart and responds by updating the visualization: filtering points, changing the axis variable, or highlighting a subset of data. A key design contribution is Eviza's handling of pragmatic ambiguity (ambiguity that arises not from the words themselves but from context and conversational implicature, which is the unstated meaning that a speaker intends a listener to infer). For example, the query "show me the expensive ones" is pragmatically ambiguous because "expensive" is relative; Eviza resolves this by asking the user to confirm its interpretation before applying the filter. This proactive disambiguation prefigures the clarification mechanisms studied in PACIFIC and CoSQL.

NL4DV. NL4DV (Natural Language for Data Visualization, Narechania et al., 2021) is a developer-facing toolkit that translates natural language queries about a tabular dataset into analytic specifications expressed in the Vega-Lite grammar (a JSON-based language for describing data visualizations). NL4DV operates through a pipeline: it identifies data attributes mentioned in the query, infers the intended analytic task (comparison, distribution, correlation, and so on), selects an appropriate chart type, and generates the corresponding Vega-Lite specification. The original NL4DV paper primarily targets one-shot utterances; conversational behaviour is typically implemented by the surrounding application by maintaining and editing prior specifications across turns.

Orko. Orko (Srinivasan & Stasko, 2018) extends the conversational natural language interface paradigm to network visualizations: graphs of nodes and edges such as social networks, citation networks, or molecule structures. Users can issue natural language commands to filter nodes, highlight paths, change the layout, or query properties of the network. Orko is relevant to this survey because its underlying technical challenges (tracking conversational context over a structured visual artifact, resolving references to previously selected nodes) are structurally analogous to those in table manipulation, even though the target artifact is a network rather than a table.

6.6 Core Technical Challenges

Both sub-communities face several shared technical challenges that distinguish manipulation from querying or translating.

Tracking a mutable state. In CTabQA and CText2SQL, the table is fixed throughout the dialogue. In manipulation, the table changes with every successful operation. The system must maintain an accurate representation of the current table state, including the effects of all prior operations in the session, in order to interpret subsequent instructions correctly. This is more demanding than tracking dialogue history over a static table because the system must track both what was *said* and what was *done*.

Mapping natural language to an action vocabulary. A user's instruction must be mapped to a specific sequence of operations drawn from the application's action library. This mapping is harder than SQL generation because the action vocabulary is application-specific: the operations available in Microsoft Excel differ from those in Google Sheets and from those in a Python pandas data frame. Systems that generate raw code (Python or VBA scripts) gain generality but sacrifice safety: executing arbitrary generated code can corrupt or delete data. Systems that use a pre-defined action library are safer but require the action library to be comprehensive enough to express all intended operations.

Instruction ambiguity and underspecification. A natural language instruction such as "clean up this table" is radically underspecified: it does not say which cells to modify, what formatting to apply, or what constitutes "clean." This underspecification is more severe than the ambiguity encountered in question answering, because the space of possible interpretations is much larger. Most current systems resolve underspecification by making a default assumption and proceeding, which works for simple tasks but fails for complex ones where the default assumption is wrong.

Verifying the result. After executing an operation, the system should verify that the resulting table matches the user's intent before proceeding to the next step. This is particularly important for destructive operations (e.g., deleting rows or overwriting formulas), where an incorrect action can cause irreversible data loss. Existing benchmarks partially capture this distinction by reporting both executability (Exec@1) and functional correctness (Pass@1), but most systems still rely on automated comparators or self-reflection modules rather than explicit user-facing confirmation and verification loops for high-risk operations.

Long-horizon task planning. A complex instruction such as "create a monthly revenue summary broken down by product category" may require a dozen or more atomic operations: selecting data, creating a pivot table, formatting the result, sorting by revenue, and generating a chart. Planning this sequence correctly requires the system to maintain an explicit goal state and to track progress toward it across many intermediate steps. SheetRM (Chen et al., 2025c) was designed specifically to stress this long-horizon setting. On SheetRM, SheetAgent achieves 44.8% Pass@1 and 77.0% SubPass@1 (Chen et al., 2025b), indicating that many intermediate subtasks can be solved even though full long-horizon completion is still far from saturated.

6.7 Category-Specific Remaining Gaps

The bottlenecks in interactive manipulation are distinctive because the system acts on mutable artifacts rather than only answering questions. Current benchmarks still underrepresent the structural messiness of real spreadsheets, including merged cells, multilevel headers, hidden formulas, embedded charts, and irregular layouts. The category is also unusually dependent on safeguards: confirmation, rollback, provenance, and other protections against destructive edits are central requirements here rather than optional interface features. Evaluation therefore needs to capture not just task completion, but whether the right edits were made safely to the evolving workbook state. These are the local bottlenecks for manipulation systems; Section 8 synthesises the broader issues they share with other categories, and Section 9 distils them into the final research agenda.

7 Agentic Table Systems

The four preceding categories all assume that the user controls the granularity of each step. In CTabQA the user asks a question; in CText2SQL the user issues a query; in Interactive Manipulation the user gives an instruction. In every case the user decides what to do next. Agentic Table Systems invert this arrangement. The user states a high-level goal and the system decides autonomously how to decompose that goal, which tools to invoke, in what order, and how to handle the results of intermediate steps.

This shift from user-directed to system-directed task execution is what the term *agentic* refers to in the context of LLM-based systems. An agent in this sense is a language model that perceives inputs from its environment (the current table state, prior tool outputs, the user's goal), reasons about what action to take next, executes that action through a tool call, observes the result, and continues this loop until the goal is achieved or the system determines it cannot be achieved. This perception-action-observation loop, often called the ReAct pattern (Reasoning and Acting, Yao et al., 2023), is the architectural foundation of many agentic table systems.

Agentic systems are among the most powerful and hardest-to-evaluate categories in this survey. They are among the most powerful because they can subsume all four preceding interaction modes within a single session: a user goal such as "prepare a summary of our quarterly revenue performance" may require the agent to query tables (CTabQA), generate SQL (CText2SQL), edit a spreadsheet (Interactive Manipulation), and compile a report, all without the user issuing separate instructions for each step. They are hard to evaluate because, to our knowledge, no single benchmark currently tests all of these capabilities within a unified conversational framework, and because the open-ended nature of the task makes it difficult to define what a correct response looks like.

We organise this section as follows. Section 7.1 describes the general-purpose agentic infrastructure that table systems build on. Section 7.2 surveys the main agentic table systems and benchmarks. Section 7.3 identifies the key design dimensions along which these systems differ. Section 7.5 concludes with category-specific remaining gaps and points forward to the cross-cutting synthesis in Section 8.

7.1 Background: Agentic LLM Infrastructure

Agentic table systems did not emerge in isolation. They build on a set of general-purpose techniques developed for LLM-based agents that are worth describing briefly before turning to the table-specific systems.

Chain-of-thought prompting (Wei et al., 2022) demonstrated that instructing a language model to produce intermediate reasoning steps before its final answer substantially improves performance on multistep tasks. Rather than asking a model to produce an answer directly, chain-of-thought prompts ask the model to think step by step, producing a reasoning trace that is visible and inspectable. In agentic table systems, chain-of-thought reasoning is used to decompose a complex goal into sub-tasks and to decide which tool to call at each step.

Retrieval-augmented generation (RAG) (Lewis et al., 2020) augments a language model with a retrieval component that fetches relevant documents or data before the model generates its response. In the table context, RAG enables agents to handle data collections that are too large to fit in a single model input: rather than encoding all available tables at once, the agent retrieves the specific tables or rows relevant to the current step of its task. TableRAG (Chen et al., 2024) applies this pattern specifically to large-scale table understanding.

Tool-augmented reasoning (Schick et al., 2023) establishes that language models can be trained or prompted to invoke external tools (calculators, search engines, code interpreters, database connectors) at appropriate points in their reasoning, rather than attempting to perform all computation internally. This is critical for table tasks because arithmetic, SQL execution, and spreadsheet operations are all better handled by deterministic interpreters than by a language model generating numbers directly. **Multiagent frameworks** (Hong et al., 2024) (exemplified by MetaGPT, a software-engineering multi-agent framework cited here for its general architectural pattern rather than as a table AI system) decompose complex tasks among multiple LLM agents, each specialised for a sub-task, that communicate through a shared message-

Table 9: Representative agentic artifacts surveyed in this section. Artifact kind: Sys = system/framework, Proto = prototype, Bmk = benchmark. Agent setup: S = single-agent, M = multi-agent, – = not applicable. Setting: DB = relational/database-native tasks, TR = general table reasoning / long-table QA, EHR = multi-tabular electronic health records, HT = heterogeneous data-analysis sources. Tool use indicates whether the artifact invokes external tools such as code execution, SQL engines, or retrieval/external operations. Eval.: B = automated benchmark, H = human evaluation, D = scenario demonstration.

Artifact	Year	Kind	Agents	Setting	Tools	Eval.	Primary Contribution
Data-Copilot (Zhang et al., 2023b)	2024	Sys	S	HT	Yes	B	Code-centric autonomous data analysis with offline interface discovery and workflow deployment
AutoTQA (Zhu et al., 2024)	2024	Sys	M	DB	Yes	B	Multi-agent tabular QA across multiple tables and database systems
Table-Critic (Yu et al., 2025)	2025	Sys	M	TR	No	B	Collaborative criticism and iterative refinement for table reasoning
TALON (Jin et al., 2025)	2025	Sys	M	TR	Yes	B	Long-table exploration and QA with planning, tool, and critic agents
DA-Code (Huang et al., 2024b)	2024	Bmk	–	HT	Yes	B	Benchmark for agent-based data-science code generation
InfiAgent-DABench (Hu et al., 2024)	2024	Bmk	–	HT	Yes	B	Benchmark for end-to-end data-analysis agents with execution feedback
Harmonia (Santos et al., 2025)	2025	Proto	S	HT	Yes	D	Interactive data harmonization with LLM reasoning and integration primitives
D-Bot (Zhou et al., 2024)	2024	Sys	M	DB	Yes	B,H	Multi-agent database diagnosis with tool use and cross-review
ChatDB (Hu et al., 2023)	2023	Sys	S	DB	Yes	B	SQL databases as symbolic memory for LLM reasoning
EHRAgent (Shi et al., 2024)	2024	Sys	S	EHR	Yes	B	Code-driven agent for few-shot complex multi-tabular EHR reasoning

passing protocol. In table settings this decomposition often follows natural functional boundaries: one agent plans the overall workflow, a second writes and executes queries, a third verifies results, and a fourth synthesises the final output.

7.2 Systems and Benchmarks

Table 9 summarises representative agentic systems, prototypes, and benchmarks discussed in this section.

Data-Copilot. Data-Copilot (Zhang et al., 2023b) presents one of the earliest end-to-end agentic systems for heterogeneous data analysis. Its central contribution is a *workflow interface*: rather than issuing individual tool calls in response to each user utterance, the agent first synthesises a complete workflow (a directed sequence of data operations and their dependencies) from the user's high-level goal, and then executes the workflow step by step. This workflow-first design separates planning from execution, allowing the agent to reason about the entire task before committing to any individual action. Data-Copilot targets heterogeneous data analysis environments rather than a single table format. In its released instantiation, it is built around large-scale financial data sources such as stocks, funds, macroeconomic indicators, and news, while supporting tasks such as querying, analysis, prediction, and visualization.

AutoTQA. AutoTQA (Zhu et al., 2024) addresses autonomous tabular question answering through a multiagent architecture in which different agents specialise in different aspects of the QA pipeline. Published in PVLDB 2024, it represents the convergence of the CTabQA and agentic paradigms: the task is still question answering, but the system decomposes it autonomously rather than processing each question as a standalone input. Its multiagent design operates across multiple tables from different database systems, allowing AutoTQA to handle questions that require several retrieval and reasoning steps without requiring the user to structure the question as a series of sub-questions.

Table-Critic. Table-Critic (Yu et al., 2025) introduces a critic agent whose role is to evaluate the answers produced by a primary QA agent and to request revisions when it detects errors or inconsistencies. This critic-reviser pattern (also called self-refinement) is an increasingly common design in LLM-based agents: rather than accepting the first output generated, the system runs one or more rounds of internal critique before presenting a result to the user. Table-Critic demonstrates that iterative critique and refinement can improve performance on standard table reasoning benchmarks such as WikiTableQuestions and TabFact.

TALON. TALON (Jin et al., 2025) is a multi-agent framework for question answering over long tables. It decomposes long-table QA into planning, tool-based table interaction, and critique, and is evaluated on long-table benchmarks derived from WikiTableQuestions and BIRD-SQL.

DA-Code, DS-STAR, and InfiAgent-DABench. DA-Code (Huang et al., 2024b) and InfiAgent-DABench (Hu et al., 2024) are benchmarks rather than systems, and they address a gap in the evaluation of agentic table capabilities.

DA-Code presents a benchmark of data science coding tasks drawn from real-world data analysis workflows, including data wrangling (the process of cleaning, reshaping, and transforming raw data into a format suitable for analysis), exploratory data analysis (EDA, the process of visually and statistically summarising a dataset to understand its structure and identify patterns), and model training. Each task requires the agent to write and execute Python code that produces a specified output from a given dataset.

DS-STAR (Nam et al., 2025) is an example of the next wave of systems evaluated on this benchmark family. It frames data-science assistance as an iterative planning-and-execution problem over heterogeneous files and open-ended analytical requests, rather than as direct code generation from a single prompt. In the cited paper, DS-STAR improves over prior DA-Code baselines by combining specialised agents for planning, routing, execution, and verification, illustrating how quickly the agentic table literature is moving from benchmark construction to full system design.

InfiAgent-DABench focuses specifically on the data analysis sub-task within agentic workflows. It evaluates agents on tasks drawn from realistic data analysis scenarios, including generating statistical summaries, identifying anomalies (data points that differ markedly from the rest of the dataset), and producing visualizations. Unlike most table benchmarks, InfiAgent-DABench measures whether the agent's generated code produces correct numerical results when executed, rather than measuring string similarity between the generated code and a reference solution.

Harmonia. Harmonia (Santos et al., 2025) addresses a specific and underexplored agentic task: conversational data harmonization. Data harmonization is the process of combining data from multiple sources that use different conventions, formats, or schemas into a single consistent dataset. For example, combining patient records from two hospitals where one records age as a number and the other records it as a date-of-birth string requires the agent to detect the inconsistency, decide how to reconcile it, and apply the transformation consistently across all records. Harmonia frames this as a conversational task in which the agent asks the user for guidance when the correct harmonization decision is ambiguous, rather than making all decisions autonomously. This conversational-agentic hybrid is an important design pattern: fully autonomous agents will make mistakes on ambiguous harmonization decisions, while systems that require the user to specify every rule become as burdensome as manual data cleaning.

D-Bot. D-Bot (Zhou et al., 2024) applies agentic LLM reasoning to database diagnosis: the task of identifying the root cause of performance problems or errors in a running database system. Published in PVLDB

2024, D-Bot demonstrates that agentic table interaction extends beyond data retrieval and manipulation to database administration. The agent issues diagnostic SQL queries, reads execution plans (structured descriptions of how a database will execute a query, used to identify performance bottlenecks), interprets log files, and proposes remediation actions. D-Bot extends the scope of what "conversational interaction with structured data" means: the user is not an analyst exploring data but a database administrator diagnosing a system, and the tables being queried are system metadata tables rather than application data tables.

ChatDB. ChatDB (Hu et al., 2023) proposes a different framing of the LLM-database relationship. Rather than treating the database as a data source to be queried, ChatDB treats it as a *symbolic memory* for the LLM. The agent writes structured facts from its reasoning process into database tables, and retrieves them in later steps using SQL. This architecture addresses the well-known limitation of language models that they cannot reliably store and recall large amounts of structured information across a long context window: by externalising memory to a database, the agent can maintain accurate structured state across an arbitrarily long session. ChatDB is conceptually close to the dialogue state tracking ideas discussed in Section 5.2, but at a much larger scale of complexity.

EHRAgent. EHRAgent (Shi et al., 2024) adapts the agentic paradigm to electronic health records, where answering a clinical question often requires reasoning across multiple linked tables. Its key idea is to let the agent generate and execute code iteratively, using execution feedback and accumulated case memory to improve multi-tabular EHR reasoning.

7.3 Key Design Dimensions

The systems described above differ along several dimensions that are worth making explicit, because these dimensions define the design space for future agentic table systems.

Single-agent versus multiagent. Some systems use a single LLM that handles all aspects of the task within one reasoning loop. Others decompose the task among multiple specialised agents. Multiagent designs can be more robust because each agent can be optimised for its sub-task, but they are more complex to coordinate and debug. The critic-reviser pattern used in Table-Critic and the planner-executor pattern used in AutoTQA represent two common multiagent decompositions. Data-Copilot, by contrast, is better viewed as a single-agent system with a plan-and-dispatch workflow.

Plan-then-execute versus step-by-step. Some systems synthesise a complete plan before executing any action (Data-Copilot). Others interleave reasoning and execution, deciding the next action only after observing the result of the previous one (the ReAct pattern). Plan-then-execute is more efficient when the goal is well-specified but fails when the plan cannot anticipate intermediate results. Step-by-step execution is more adaptive but can get stuck in loops or lose track of the overall goal in long sessions.

Tool repertoire. Systems vary in which tools they can invoke: SQL execution engines, Python code interpreters, spreadsheet APIs, web search, and file I/O. A richer tool repertoire enables more tasks but increases the difficulty of tool selection (deciding which tool to use at each step) and tool composition (combining multiple tools correctly in a single workflow).

Conversational integration. Some agentic systems are designed to operate entirely autonomously, without user interaction during execution. Others integrate conversational turns into the execution loop, pausing to ask for clarification when the goal is ambiguous or when an intermediate result is unexpected. Harmonia is the clearest example of conversational-agentic integration; most other systems in this category are predominantly autonomous. This dimension is the primary connection between the agentic category and the conversational themes that unite this survey.

Data heterogeneity. Some systems operate on a single structured data format (relational database, spreadsheet). Others handle heterogeneous data: combining structured tables with unstructured text, time-series data, images, or external API data. Data-Copilot and InfiAgent-DABench represent the heterogeneous end of this spectrum; D-Bot and AutoTQA represent the structured-only end.

7.4 Relationship to the Preceding Categories

Agentic table systems do not replace the preceding four categories; they build on them. The taxonomy survey by Tian et al. (2026) identifies five competencies that a complete table agent must possess: structure understanding (knowing how to encode and interpret tables), semantic understanding (knowing what a user's goal means), retrieval and compression (knowing how to find the relevant data within a large collection), executable reasoning (knowing how to produce and execute correct code or SQL), and cross-domain generalization (performing reliably on tables from domains not seen during training). Each of these competencies maps onto one or more of the preceding categories: structure understanding maps to Foundations, semantic understanding maps to CTabQA, executable reasoning maps to CText2SQL and Interactive Manipulation, and retrieval and compression maps to techniques used across all four categories.

What the agentic category adds is *orchestration*: the ability to invoke these competencies in sequence, coordinate their outputs, and maintain coherence across a session that may involve dozens of steps. In the current literature, no system appears to achieve all five competencies reliably across diverse real-world data environments, and the evaluation frameworks needed to measure all five simultaneously do not yet exist.

7.5 Category-Specific Remaining Gaps

Agentic table systems face a distinctive evidence problem: they perform long tool-using workflows, but are rarely evaluated under sustained conversational supervision. Most benchmarks still test task completion more than interactive planning quality, intermediate recovery, or the visibility of provenance across generated code, SQL, search, and edits. The category also remains closer to autonomous batch execution than to true back-and-forth analytical partnership, so user intervention, correction, and mixed-initiative control are still underdeveloped. In this category especially, trustworthy orchestration is part of the core task, not an afterthought. These are the local bottlenecks for agentic systems; the recurring cross-category patterns are synthesised in Section 8 and condensed in Section 9.

8 Cross-Cutting Themes

The five categories surveyed in the preceding sections address different interaction modes and different data formats, but they face a common set of challenges that do not belong to any one category. These challenges arise wherever a system must sustain a coherent interaction with a user over multiple turns of a conversation grounded in structured data. We identify three such cross-cutting themes: intent disambiguation and clarification (Section 8.1), dialogue context tracking (Section 8.2), and evaluation and benchmarks (Section 8.3). Together they explain why otherwise different categories often fail in the same ways.

Understanding these themes together, rather than within individual category sections, serves two purposes. First, it reveals structural parallels that are invisible when each category is treated in isolation: the way PACIFIC handles clarification in CTabQA and the way CoSQL handles clarification in CText2SQL are conceptually the same mechanism applied in different settings, yet the two research communities have usually been discussed and evaluated separately. Second, it identifies the gaps that recur across otherwise fragmented literatures, allowing this section to synthesise them once before Section 9 turns that synthesis into a concise final agenda.

8.1 Intent Disambiguation and Clarification

Every conversational system grounded in structured data must at some point encounter a user input whose meaning is not uniquely determined by the available context. The input may be ambiguous because the user used a word that maps to multiple database columns, because the user omitted information that the system needs to generate a correct answer, or because the user's goal could be satisfied in more than one way and the user has not indicated a preference. The question of what a system should do in this situation is the clarification problem.

8.1.1 Forms of Ambiguity in Table Interactions

Ambiguity in table-grounded dialogue takes several distinct forms, each requiring a different resolution strategy.

Lexical ambiguity arises when a word or phrase in the user's input maps to more than one column, table, or value in the data. A user who asks "what are the revenues?" over a database that contains both gross revenue and net revenue columns has issued a lexically ambiguous question. The system must either choose one interpretation, ask the user to specify, or return both.

Referential ambiguity arises when a pronoun or demonstrative expression could plausibly refer to more than one entity introduced in the prior dialogue. A question such as "how did it change year over year?" is referentially ambiguous if the prior turns introduced multiple numerical quantities.

Scope ambiguity arises when the extent of a user's request is unclear. A question such as "show me all the recent transactions" is scope-ambiguous because "recent" is not defined: it could mean the last day, the last month, or the last year. Scope ambiguity is particularly common in financial and scientific table interactions where temporal and numerical ranges are frequently left implicit.

Underspecification is a more radical form of ambiguity in which the user's instruction is compatible with a very large number of different actions. As discussed in Section 6.6, an instruction such as "clean up this table" is so underspecified that no single correct interpretation exists. Underspecification is most common in Interactive Table Manipulation.

8.1.2 Current Approaches to Clarification

The research literature on clarification in conversational table AI is dominated by two established approaches, always-answer and clarification question generation, plus a smaller line of exploratory work on uncertainty-aware response presentation.

Always-answer systems commit to the most likely interpretation of an ambiguous input and produce an answer without flagging the ambiguity. This is the default behaviour of most CTabQA and CText2SQL systems. It performs well on unambiguous inputs and on inputs where one interpretation is strongly dominant, but degrades on inputs where two or more interpretations are roughly equally likely, because the system will be wrong half the time and will not signal its uncertainty to the user.

Clarification question generation (CQG) systems aim to detect ambiguity and respond by generating a natural language question that prompts the user to resolve it. (Deng et al., 2022) introduce PACIFIC, a conversational table QA dataset that incorporates clarification interactions within multi-turn reasoning. (Yu et al., 2019a) present CoSQL, which includes clarification-like exchanges as part of a conversational Text-to-SQL framework. (Zhong et al., 2021) propose NL-EDIT, which addresses a related problem by enabling users to correct misgenerated SQL queries through natural language feedback, requiring the system to produce targeted clarifications that guide error resolution.

The central challenge of CQG is deciding *when* to ask. A system that asks for clarification on every mildly ambiguous input becomes tedious; a system that asks too rarely makes systematic errors. Deng et al. (2022) report a persistent gap to human performance on the clarification decision, suggesting that reliably estimating intent uncertainty remains challenging for current models. The broader survey by Deng et al. (2023) places this challenge in the context of proactive dialogue systems more generally, covering not only clarification but also target-guided dialogue (in which the system steers the conversation toward a pre-specified goal) and non-collaborative settings.

Uncertainty-aware response presentation is better viewed here as an early proposal than as a mature third paradigm. In this framing, the system produces multiple candidate answers (or SQL queries) corresponding to different interpretations, ranks them by estimated probability, and presents the top-ranked answer alongside an indication of its confidence. ABG-CoQA (Guo et al., 2021) (an arXiv preprint; we are not aware of a widely used reproduction in the table-centric setting) studies ambiguity in conversational QA and argues that presenting multiple candidate answers with associated confidence scores can be preferable

to committing to a single interpretation or always asking a clarification question, because it lets the user resolve ambiguity directly.

8.1.3 Cross-Category Synthesis

Taken together, the literature shows three shared weaknesses in clarification. First, explicit evaluation of clarification behaviour is rare outside PACIFIC and CoSQL, so the field still lacks a stable way to compare systems on when they ask, when they answer, and how well they handle ambiguity. Second, the threshold for asking versus answering is inconsistent across datasets and system designs, which makes clarification performance difficult to interpret even when it is reported. Third, the user's response to a clarification question is seldom modelled as a structured update to dialogue state; instead, it is often treated as just another utterance, which leaves systems vulnerable to repeating the same ambiguity or propagating the wrong assumption forward.

8.2 Dialogue Context Tracking

Every multiturn system must solve the problem of maintaining an accurate and useful representation of the conversational context accumulated across prior turns. Without this representation, the system cannot resolve coreferences, interpret context-dependent questions, or avoid repeating information already established in the dialogue. The form that this representation takes differs substantially across the five categories, but the underlying requirement is the same.

8.2.1 What Context Tracking Means in Each Category

In **CTabQA**, context tracking means maintaining a record of which entities, values, and columns have been discussed, what numerical results have been computed, and what the user appears to be trying to understand. The cTBLS system (Sundar & Heck, 2023) represents this through the conversation history and the retrieved table cells; ConvFinQA (Chen et al., 2022) represents it through a chain of numerical reasoning steps that carry intermediate results forward.

In **CText2SQL**, context tracking takes the specific form of SQL context: the current SQL query encodes which tables are selected, which conditions are active, which aggregations are in scope, and which columns are being returned. Systems such as IST-SQL (Wang et al., 2021) maintain an explicit interaction state that records this SQL context and updates it at each turn, while RASAT (Zhong et al., 2022) encodes it implicitly through coreference relation types in the attention mechanism. HIE-SQL (Zheng et al., 2022) uses a history-enhanced encoder that directly attends over prior question and SQL pairs to resolve cross-turn dependencies.

In **Interactive Table Manipulation**, context tracking must follow not only what was said but also what was done. The table state changes with every executed operation, and subsequent instructions must be interpreted relative to the current table, not the original one. This mutable-state tracking is more demanding than tracking dialogue over a static table because the context the system must maintain includes the full history of table modifications, not just the history of utterances.

In **Agentic Table Systems**, context tracking expands further to include the outputs of all tool calls, the results of intermediate code executions, and the current state of all data artifacts that the agent has produced or modified during the session. ChatDB (Hu et al., 2023) externalises this context to a database, making it queryable and persistent across an arbitrarily long session.

8.2.2 The Dialogue State Tracking Analogy

The challenge of context tracking in conversational table AI is structurally analogous to dialogue state tracking (DST) in task-oriented dialogue systems, a well-studied problem in the spoken dialogue and conversational AI literature (Ji et al., 2020; Li et al., 2020). In standard DST, the state is a set of slot-value pairs that represent the user's current goal (for example, in a hotel booking dialogue, the slots might be location, check-in date, price range, and room type). The state is updated at each turn as the user provides more information or changes their preferences.

In conversational table AI, the analogous state is richer and less structured. Rather than a flat set of slot-value pairs, the state includes the current SQL query (in CText2SQL), the history of retrieved cells and computed values (in CTabQA), the current table state (in Interactive Manipulation), and the current workflow execution state (in Agentic Systems). This richness is both an opportunity and a challenge: the richer state enables more complex interactions, but it also makes state maintenance harder and makes errors more difficult to detect and correct.

Despite this structural parallel, the DST and conversational table AI literatures have developed largely in parallel without substantial cross-fertilisation. Within CText2SQL specifically, this analogy also helps organise the model families surveyed in Section 5: SQL-editing methods approximate state updates, rewriting methods externalise context recovery, and IST-SQL (Wang et al., 2021) comes closest to an explicit state model. Question rewriting (Vakulenko et al., 2021), which converts a context-dependent question into a self-contained one by explicitly incorporating the relevant context, therefore acts as a concrete bridge between the two traditions. Integrating the principled state maintenance techniques developed for task-oriented DST with the richer state representations required for table interactions is a promising and underexplored research direction.

8.2.3 Cross-Category Synthesis

Across categories, context tracking remains brittle for three related reasons. First, most systems still rely on raw dialogue history rather than a structured, queryable summary of what remains in scope, what has already been computed, and which assumptions have been confirmed. Second, performance degrades with dialogue length wherever it has been measured (Chen et al., 2022), which suggests that current representations do not scale gracefully as state accumulates. Third, error recovery is largely absent: once a mistaken assumption enters the dialogue state, later turns usually inherit it rather than challenge, repair, or replace it.

8.3 Evaluation and Benchmarks

Evaluation is the most fragmented aspect of conversational table AI. The five categories use different metrics, different benchmark formats, and different notions of what constitutes a correct response. This fragmentation makes it impossible to compare progress across categories, to identify which capabilities are improving fastest, or to determine whether a new method represents a genuine advance across the field or only an improvement on a narrow benchmark.

8.3.1 The Benchmark Landscape

Table 10 provides a structured view of the benchmark landscape in two panels. Panel A collects the core multiturn and interactive benchmarks that anchor the survey taxonomy. Panel B isolates adjacent single-turn predecessors and background datasets that matter historically, but should not be overread as a directly comparable benchmark family with Panel A.

Table 11 deliberately separates *reporting conditions* from *performance values*. The point is not to rank benchmarks or to suggest a single cross-category leaderboard. It should not be used to infer that one category is stronger or more mature than another from one headline number. It is to show where public evaluation artifacts differ in split visibility, metric family, and task definition, because those differences are precisely what make score-level comparisons fragile across categories.

This separation is deliberate. Table 10 summarises the benchmark landscape itself. Table 11 records the public reporting conditions under which scores are exposed. Once the task definition, split, and public artifact differ, a single cross-category "best number" can imply comparability that the underlying benchmarks do not actually support.

Several observations emerge from this reporting-context view. The benchmark landscape is dominated by the finance and Wikipedia domains. Office-focused benchmarks exist, but they typically evaluate scripted multi-step tasks rather than open-ended, user-driven dialogue with clarification and context repair. Agentic benchmarks are usually single-turn at the level of evaluation instances even though the systems themselves

Table 10: Benchmark landscape across the five taxonomy categories, presented in two panels to separate the core survey benchmarks from adjacent background predecessors. Category abbreviations: F = Foundations background, CQ = CTabQA, CS = CText2SQL, M = Interactive Manipulation, A = Agentic. Multiturn indicates context-dependent turns. Hybrid indicates table plus prose evidence. Primary metric abbreviations are defined in Section 8.3.2. Size is reported as number of dialogues or question sequences where available, and as number of individual QA pairs otherwise.

Dataset	Cat.	Year	Multiturn	Domain	Hybrid	Size	Primary Metric
Panel A. Core multiturn and interactive benchmarks in the survey taxonomy							
WikiTableQ (Pasupat & Liang, 2015)	F	2015	No	Wikipedia	No	22K pairs	FM Acc.
SQA (Iyyer et al., 2017)	CQ	2017	Yes	Wikipedia	No	6,066 seq.	EM
SParC (Yu et al., 2019b)	CS	2019	Yes	138 domains	No	4,298 seq.	QEM, IEM
CoSQL (Yu et al., 2019a)	CS	2019	Yes	138 domains	No	3,007 dial.	EX, CQG, DA
HybriDialogue (Nakamura et al., 2022)	CQ	2022	Yes	Wikipedia	Yes	4.8k dial.	ROUGE, EM
PACIFIC (Deng et al., 2022)	CQ	2022	Yes	Finance	Yes	2,757 dial.	EM, CQG F1
ConvFinQA (Chen et al., 2022)	CQ	2022	Yes	Finance	Yes	3,892 dial.	Exec. acc.
MMCoQA (Li et al., 2022)	CQ	2022	Yes	General	Yes	1K+ dial.	EM, F1
SheetRM (Chen et al., 2025c)	M	2024	Yes	Office	No	varies	SubPass@1
SpreadsheetBench (Ma et al., 2024)	M	2024	Yes	Office	No	varies	Pass@1
InfiAgent-DABench (Hu et al., 2024)	A	2024	No	General	No	varies	Acc. by Qs.
DA-Code (Huang et al., 2024b)	A	2024	No	General	No	varies	Total acc.
iTBLS (Sundar et al., 2025)	CQ	2025	Yes	Scientific	No	4K+ dial.	EM, BERTScore
Panel B. Background single-turn or predecessor benchmarks *These rows are included for lineage and context only. They are not intended to define a single directly comparable benchmark family with Panel A.*							
TAT-QA (Zhu et al., 2021)	CQ	2021	No	Finance	Yes	16K pairs	EM, F1
FinQA (Chen et al., 2021b)	CQ	2021	No	Finance	Yes	8,281 pairs	Exec. acc.
BIRD (Li et al., 2023b)	CS	2023	No	95 databases (37 domains)	No	12K pairs	EX

are multistep. Across categories, the reported metrics remain largely incomparable, which is precisely why a transparent statement of reporting conditions is more defensible than a single mixed leaderboard.

8.3.2 Evaluation Metrics and Their Limitations

Table 12 summarises the primary evaluation metrics that recur across the benchmark landscape and reporting conditions above, together with their scope and known limitations.

Table 11: Public reporting conditions for selected core benchmarks. This is intentionally *not* a leaderboard table and must not be read as evidence of cross-category superiority. It records how results are usually reported, where they are exposed publicly, and what makes direct score comparisons difficult across categories.

Benchmark	**Cat.**	**Common public metric(s)**	**Typical public artifact**	**Split visibility**	**Main comparability caveat**
WikiTableQuestions	F	Fuzzy Match accuracy	Paper and public benchmark reporting	Test	Important historical predecessor, but not a conversational benchmark
SQA	CQ	EM	Paper tables	Test	Decomposed sequential QA over Wikipedia tables; no explicit clarification task
HybriDialogue	CQ	ROUGE, EM	Paper tables	Test	Hybrid retrieval and generation setting; oracle-style retrieval variants also appear in the literature
PACIFIC	CQ	EM, clarification-question metrics	Paper tables	Test	Mixes answer quality with clarification quality within one benchmark
ConvFinQA	CQ	Execution accuracy	Paper tables	Dev and test reporting both appear in later work	Program-generation setting over hybrid financial documents, not plain span extraction
SParC	CS	QM, IM, execution-family metrics	Hidden-test leaderboard and paper tables	Dev and hidden test	Different papers emphasize different metric families, so exact-match and execution results should not be collapsed into one scalar
CoSQL	CS	EX, question / interaction metrics, dialogue-act and clarification metrics	Hidden-test leaderboard and paper tables	Dev and hidden test	Multitask benchmark; SQL generation, answer quality, and dialogue behavior are reported separately
SheetRM	M	Pass@1, Sub-Pass@1	Paper tables	Test	Measures long action sequences over spreadsheets rather than open-ended free-form dialogue quality
SpreadsheetBench	M	Pass@1	Public leaderboard and paper	Public leaderboard	Execution correctness over workbook states is central, which is not directly comparable to QA or SQL metrics
InfiAgent-DABench	A	Aggregate task accuracy	Project page and paper	Public benchmark reporting	Aggregates heterogeneous agentic tasks rather than a single dialogue objective
DA-Code	A	Aggregate task accuracy	Paper tables	Paper-reported splits	Mixes data-wrangling, machine-learning, and EDA subtasks in one evaluation bundle

The limitations enumerated in Table 12 point to a common underlying problem: every current metric measures a proxy for what we actually care about. What we care about is whether the system correctly understood what the user wanted, correctly identified the relevant data, and correctly produced a response that satisfies the user's goal. No current metric measures all three of these components simultaneously, and none measures them across a full multiturn dialogue without decomposing the dialogue into individual turns first.

8.3.3 Fragmentation and Its Consequences

The evaluation fragmentation documented above has three practical consequences for the field.

First, it makes it difficult to identify genuine progress. A paper that reports a 3% improvement in QEM on SParC is not obviously comparable to a paper that reports a 5% improvement in Pass@1 on SpreadsheetBench. Both may represent equivalent amounts of scientific progress, or very different amounts. Without a shared metric, there is no way to know.

Table 12: Evaluation metrics that recur across the benchmark landscape, with their primary use cases and known limitations.

Metric	Used In	What It Measures	Key Limitation
FM Acc. (Fuzzy Match)	Foundations (WTQ)	Approximate string match between prediction and gold answer under token/substring normalization	Can over-credit partially correct spans; sensitive to normalization heuristics and answer formatting
EM (Exact Match)	CTabQA	Whether predicted answer exactly matches the gold answer after normalization	Penalises paraphrases/synonyms; brittle to formatting and minor wording differences
EX (Execution Accuracy)	CText2SQL	Whether predicted SQL yields the same result set as the gold query when executed	Can be fooled by spurious queries that coincidentally return the same result; does not assess reasoning fidelity
Exec. acc. (program)	CTabQA (programmatic / tool-use QA)	Whether generated program/code (or intermediate tool outputs) executes to the correct final answer	Execution can be correct for the wrong reasons; depends on environment/tooling assumptions and error handling
ROUGE-1	CTabQA (generative responses)	Unigram overlap between generated response and reference response	Weak proxy for factual correctness; can reward fluent but incorrect answers and punish valid paraphrases
F1	CTabQA	Token-level overlap between prediction and gold answer (harmonic mean of precision/recall)	Still surface-form dependent; partial credit may not correspond to semantic correctness
SubPass@1	Interactive Manipulation	Fraction of sub-tasks completed correctly on the first attempt	Sub-task definitions are benchmark-specific; not directly comparable across different manipulation suites
Pass@1	Interactive Manipulation	Whether the full task completes correctly on the first attempt	Binary and coarse; does not distinguish near-misses from failures or measure efficiency/interaction cost
Accuracy by Questions	Agentic	Fraction of questions/tasks solved correctly (question-level accuracy)	Does not evaluate tool-call efficiency, reasoning trace quality, or robustness; can hide per-category failure modes
Total acc. (aggregate)	Agentic	Aggregate accuracy over multiple agent task types (e.g., DW/ML/EDA) under a benchmark's scoring protocol	Aggregation weights and task mix are benchmark-defined; overall score can mask weaknesses on critical subtasks

Second, it creates incentives to optimise for benchmark metrics at the expense of capabilities that are not measured. Systems trained to maximise execution accuracy on CoSQL, for example, have limited incentive to improve their clarification behaviour, because such behaviour is not directly or explicitly rewarded by the evaluation metrics, even though it is a genuine component of the task. The survey by Sui et al. (2024a) systematically evaluates LLM capabilities on structured table data and demonstrates that models which score well on standard benchmarks often fail on structural reasoning tasks that require understanding the organisation and layout of a table rather than just its content. StructBench (Gu et al., 2025) similarly shows that current LLMs struggle with reasoning over structure-rich inputs, even when they perform strongly on content-oriented tasks.

Third, it prevents cumulative scientific progress. A new CTabQA system that achieves the best published result on HybriDialogue cannot be compared to a new CText2SQL system that achieves the best result on SParC, because the two tasks, the two datasets, and the two metrics are all different. Building a shared evaluation framework that spans at least a subset of the five categories is the most important infrastructure investment the conversational table AI community could make.

8.3.4 A Note on Multilinguality

The benchmark landscape surveyed here is almost entirely English. Beyond Arabic variants and transfer studies building on SParC (Yu et al., 2019b) and cross-lingual transfer studies in CText2SQL (Liu et al., 2022a), multilingual conversational table AI is a largely unexplored area. This gap is consequential: enterprise database users worldwide issue queries in their native languages, and the structural challenges of multiturn SQL generation or table QA compound with the linguistic challenges of cross-lingual transfer. No benchmark currently tests conversational table AI in any non-English language with a multiturn protocol. Constructing multilingual CText2SQL and CTabQA datasets, particularly for low-resource languages, is an open contribution of practical importance.

8.3.5 Towards Better Evaluation

Three directions for improving evaluation are identifiable from the current literature.

Process evaluation alongside outcome evaluation. Current metrics measure the final output of a system but not the process by which it was produced. A system that generates a correct answer by accidentally retrieving the right cell has no more claim to having understood the question than one that retrieved the wrong cell and guessed. Evaluating the evidence cited by a system, the intermediate reasoning steps it produced, and the confidence it expressed alongside its answer would provide a much richer picture of system capability than final-answer accuracy alone. The Table Meets LLM benchmark (Sui et al., 2024a) and TabPedia (Zhao et al., 2024) represent steps toward richer evaluation of table understanding capabilities, though neither is specifically conversational.

Dialogue-level metrics. All current metrics evaluate individual turns in isolation. A metric that measures the coherence of a full dialogue, the degree to which the system's responses build on each other correctly across turns, the rate at which context errors propagate, and the quality of clarification exchanges would be more informative than any per-turn metric. Interaction-level exact match (IEM) is a crude step in this direction, but its binary all-or-nothing design makes it too strict to be informative in practice.

Cross-category benchmarks. No benchmark currently tests a system on tasks that span multiple categories, for example a task that requires answering a question (CTabQA), then editing the table based on that answer (Interactive Manipulation), and then generating a SQL query to verify the edit (CText2SQL). Such a benchmark would test the integrated capability that end-to-end conversational table AI requires, and its absence is the most significant structural gap in the current evaluation landscape.

9 Open Problems and Future Directions

Section 8 synthesised the recurring limitations that appear across CTabQA, CText2SQL, interactive manipulation, and agentic table systems. This section does not restate those issues category by category. Instead, it condenses them into six research problems whose resolution would most improve the field as a whole. These six problems absorb the recurring themes discussed earlier and therefore serve as the paper's single forward-looking agenda.

9.1 Unified Evaluation and Benchmark Design

The most consequential structural gap in the field is still the absence of a shared evaluation framework. As Section 8.3 showed, categories reuse familiar metrics such as EM, execution accuracy, or Pass@1, but they instantiate them under different task definitions, annotation protocols, and test distributions, so scores are rarely comparable across categories in a defensible way. A useful next step is not one more category-specific

leaderboard, but a benchmark suite that evaluates a common set of systems on tasks drawn from multiple categories under a shared protocol. That suite should pair outcome scoring with process annotations: which cells were used, which intermediate operations were performed, which clarification exchanges occurred, and how confidence changed across the dialogue. Without that infrastructure, the field will keep improving within silos without being able to show integrated progress. A convincing contribution here would deliver a benchmark suite, an annotation protocol, and a reporting template that multiple categories can all adopt without task-specific reinterpretation.

A related meta-research problem is to quantify the field's fragmentation directly rather than only describing it qualitatively. A practical starting point would be a bibliometric audit over the surveyed corpus: assign each paper to one primary category, construct a within- versus cross-category citation matrix, and report how citation flow changes over time. Such an analysis would not replace unified technical evaluation, but it would make cross-community isolation itself measurable and would test one of this survey's motivating observations with explicit evidence.

9.2 Dialogue State, Long-Horizon Coherence, and Error Recovery

Performance degrades with dialogue length wherever it has been measured, whether in ConvFinQA-style numerical dialogue, conversational SQL, or long operation sequences in manipulation benchmarks. The shared cause is that most systems still treat prior turns as raw history rather than as a compact, structured state capturing what remains in scope, what has already been computed, and which assumptions have been confirmed. This creates a second problem as well: once an early turn is wrong, later turns usually inherit that error rather than detect and repair it. A major advance would therefore be a dialogue-state representation that supports retrieval, revision, and recovery, rather than merely longer context windows. Progress should be visible as flatter performance curves over turn depth and explicit recovery from injected earlier-turn errors.

9.3 Clarification and Ambiguity Management

Most current systems still guess when a user's request is ambiguous. Only a small subset of table-grounded benchmarks, most notably PACIFIC and CoSQL, evaluate clarification explicitly, and even there the field lacks a general account of when a system should ask, what it should ask, and how the answer should constrain subsequent reasoning. Treating clarification as a first-class capability requires three linked components: an ambiguity detector, a minimal and targeted clarification generator, and a state update mechanism that integrates the user's response into the ongoing dialogue. Until those three pieces are studied together, clarification will remain a special case rather than a core conversational competence. A useful benchmark here would score not just final accuracy, but whether the system asked only when needed, asked the right question, and updated its internal state correctly after the reply.

9.4 Numerical, Multi-Table, and Cross-Source Reasoning

Real analytical work rarely involves one table and one step. Users chain numerical results across turns, join multiple tables, compare documents from different sources, and shift attention between evidence types without restating every dependency explicitly. Current benchmarks usually isolate only one slice of that behaviour: ConvFinQA and PACIFIC stress numerical chaining, Spider 2.0 raises single-turn multi-table difficulty, and spreadsheet or agentic benchmarks cover only fragments of cross-source reasoning. A high-leverage research direction is therefore to model intermediate quantities, table provenance, and join structure as persistent conversational state rather than temporary by-products of a single turn. That would move the field closer to the actual structure of data analysis instead of its single-table approximation. A concrete success criterion would be sustained performance on multi-table, multi-document dialogues in which intermediate quantities must be reused several turns later.

9.5 Real-World Robustness, Hybrid Documents, and Domain Transfer

The benchmark distributions surveyed in this paper remain cleaner, flatter, and more English-centric than the environments in which these systems are meant to operate. Real spreadsheets contain merged cells, irregular layouts, hidden formulas, and embedded charts; real reports combine tables with prose and figures; real deployment settings require domain transfer into finance, medicine, science, law, and low-resource languages. These are often treated as separate problems, but they express the same underlying weakness: current systems are over-adapted to narrow benchmark conventions. The field needs benchmarks and models that preserve real structural messiness while remaining safe to release, for example through anonymisation pipelines, synthetic-yet-realistic table generation, and multilingual or domain-controlled evaluation protocols. A convincing step forward would show that a model trained under these conditions degrades gracefully across domains, layouts, and languages rather than collapsing outside its home benchmark.

9.6 Trustworthy Deployment: Interpretability, Privacy, and Safe Action

A deployable conversational table system must be auditable, privacy-aware, and safe to use on mutable data. That means more than producing a correct final answer: the system should expose the evidence cells it relied on, the operations it performed, the prior-turn assumptions it used, and enough provenance for a user to verify the result. It also means respecting access control, limiting schema leakage, and preventing destructive actions from being executed without confirmation, rollback, or human oversight. These concerns span answer generation, SQL execution, spreadsheet manipulation, and agentic workflows alike. If the field solves them only after benchmark accuracy plateaus, it will delay the transition from research demos to trustworthy analytical systems. Progress here should therefore be measured with deployment-facing criteria such as provenance exposure, rollback support, permission awareness, and user-verifiable action traces, not only end-task accuracy.

10 Conclusion

What this survey makes visible is that conversational AI over structured data is not a loose collection of niche tasks, but an emerging interaction stack. Foundations methods make tables legible to language models; CTabQA and CText2SQL study how intent and context are carried across turns; interactive manipulation systems add mutable state and execution; agentic systems add multistep planning and orchestration. Seen together, these areas reveal a single broader problem: building systems that can work with structured information the way people actually do, iteratively, contextually, and with the ability to ask, act, verify, and recover. That unifying view also clarifies why progress has felt uneven. Individual categories have advanced on their own benchmarked subproblems, but the capabilities needed for real deployment do not respect category boundaries. Clarification, long-horizon context tracking, numerical faithfulness, evidence grounding, robustness to messy and hybrid tables, and evaluation under realistic user goals all cut across the taxonomy. The main bottleneck is therefore not only model quality within each category, but the absence of shared evaluation and system designs that integrate these capabilities coherently. The next stage of the field should move from isolated task optimization toward integrated conversational data systems that can query, reason, manipulate, and orchestrate over structured information in transparent and user-aligned ways. The most valuable future work will be the work that connects these capabilities rather than improving them in isolation.

References

James E Allen, Curry I Guinn, and Eric Horvtz. Mixed-initiative interaction. *IEEE Intelligent Systems and their Applications*, 14(5):14–23, 1999.

Ruisheng Cao, Lu Chen, Zhi Chen, Yanbin Zhao, Su Zhu, and Kai Yu. LGESQL: Line graph enhanced text-to-SQL model with mixed local and non-local relations. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *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), pp. 2541–2555, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.198. URL https://aclanthology.org/2021.acl-long.198/.

Bingfeng Chen, Shaobin Shi, Yongqi Luo, Boyan Xu, Ruichu Cai, and Zhifeng Hao. Track-SQL: Enhancing generative language models with dual-extractive modules for schema and context tracking in multi-turn text-to-SQL. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 10690–10708, Albuquerque, New Mexico, April 2025a. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.536. URL https://aclanthology.org/2025.naacl-long.536/.

Ran Chen, Di Weng, Yanwei Huang, Xinhuan Shu, Jiayi Zhou, Guodao Sun, and Yingcai Wu. Rigel: Transforming tabular data by declarative mapping. *IEEE Trans. Vis. Comput. Graph.*, 29(1):128–138, 2023. doi: 10.1109/TVCG.2022.3209385. URL https://doi.org/10.1109/TVCG.2022.3209385.

Si-An Chen, Lesly Miculicich, Julian Eisenschlos, Zifeng Wang, Zilong Wang, Yanfei Chen, Yasuhisa Fujii, Hsuan-Tien Lin, Chen-Yu Lee, and Tomas Pfister. Tablerag: Million-token table understanding with language models. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/88dd7aa6979e352fda7c4952ca8eac59-Abstract-Conference.html.

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

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

Wenhu Chen, Ming-Wei Chang, Eva Schlinger, William Yang Wang, and William W. Cohen. 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, 2021a. URL https://openreview.net/forum?id=MmCRswl1UYl.

Yibin Chen, Yifu Yuan, Zeyu Zhang, Yan Zheng, Jinyi Liu, Fei Ni, Jianye Hao, Hangyu Mao, and Fuzheng Zhang. Sheetagent: Towards a generalist agent for spreadsheet reasoning and manipulation via large language models. In Guodong Long, Michale Blumestein, Yi Chang, Liane Lewin-Eytan, Zi Helen Huang, and Elad Yom-Tov (eds.), *Proceedings of the ACM on Web Conference 2025, WWW 2025, Sydney, NSW, Australia, 28 April 2025- 2 May 2025*, pp. 158–177. ACM, 2025b. doi: 10.1145/3696410.3714962. URL https://doi.org/10.1145/3696410.3714962.

Yibin Chen, Yifu Yuan, Zeyu Zhang, Yan Zheng, Jinyi Liu, Fei Ni, Jianye Hao, Hangyu Mao, and Fuzheng Zhang. Sheetagent: Towards a generalist agent for spreadsheet reasoning and manipulation via large language models. In Guodong Long, Michale Blumestein, Yi Chang, Liane Lewin-Eytan, Zi Helen Huang, and Elad Yom-Tov (eds.), *Proceedings of the ACM on Web Conference 2025, WWW 2025, Sydney, NSW, Australia, 28 April 2025- 2 May 2025*, pp. 158–177. ACM, 2025c. doi: 10.1145/3696410.3714962. URL https://doi.org/10.1145/3696410.3714962.

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. FinQA: A dataset of numerical reasoning over financial data. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3697–3711, Online and Punta Cana, Dominican Republic, November 2021b. Association for

Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.300. URL https://aclanthology.org/2021.emnlp-main.300/.

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

Zhoujun Cheng, Tianbao Xie, Peng Shi, Chengzu Li, Rahul Nadkarni, Yushi Hu, Caiming Xiong, Dragomir Radev, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. Binding language models in symbolic languages. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=lH1PV42cbF.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. QuAC: Question answering in context. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2174–2184, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1241. URL https://aclanthology.org/D18-1241/.

Kevin Clark and Christopher D. Manning. Deep reinforcement learning for mention-ranking coreference models. In Jian Su, Kevin Duh, and Xavier Carreras (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2256–2262, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1245. URL https://aclanthology.org/D16-1245/.

E. F. Codd. A relational model of data for large shared data banks. *Commun. ACM*, 13(6):377–387, 1970. doi: 10.1145/362384.362685. URL https://doi.org/10.1145/362384.362685.

Deborah A. Dahl, Madeleine Bates, Michael Brown, William Fisher, Kate Hunicke-Smith, David Pallett, Christine Pao, Alexander Rudnicky, and Elizabeth Shriberg. Expanding the scope of the ATIS task: The ATIS-3 corpus. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*, 1994. URL https://aclanthology.org/H94-1010/.

Yang Deng, Wenqiang Lei, Wenxuan Zhang, Wai Lam, and Tat-Seng Chua. PACIFIC: Towards proactive conversational question answering over tabular and textual data in finance. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 6970–6984, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.469. URL https://aclanthology.org/2022.emnlp-main.469/.

Yang Deng, Wenqiang Lei, Wai Lam, and Tat-Seng Chua. A survey on proactive dialogue systems: Problems, methods, and prospects. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, pp. 6583–6591. ijcai.org, 2023. doi: 10.24963/IJCAI.2023/738. URL https://doi.org/10.24963/ijcai.2023/738.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Xi Fang, Weijie Xu, Fiona Anting Tan, Ziqing Hu, Jiani Zhang, Yanjun Qi, Srinivasan H. Sengamedu, and Christos Faloutsos. Large language models (llms) on tabular data: Prediction, generation, and understanding - A survey. *Trans. Mach. Learn. Res.*, 2024, 2024. URL https://openreview.net/forum?id=IZnrCGF9WI.

Dawei Gao, Haibin Wang, Yaliang Li, Xiuyu Sun, Yichen Qian, Bolin Ding, and Jingren Zhou. Text-to-sql empowered by large language models: A benchmark evaluation. *Proc. VLDB Endow.*, 17(5):1132–1145, 2024. doi: 10.14778/3641204.3641221. URL https://www.vldb.org/pvldb/vol17/p1132-gao.pdf.

Zhouhong Gu, Haoning Ye, Xingzhou Chen, Zeyang Zhou, Hongwei Feng, and Yanghua Xiao. StrucText-eval: Evaluating large language model's reasoning ability in structure-rich text. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 223–244, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.11. URL https://aclanthology.org/2025.acl-long.11/.

Meiqi Guo, Mingda Zhang, Siva Reddy, and Malihe Alikhani. Abg-coqa: Clarifying ambiguity in conversational question answering. In Danqi Chen, Jonathan Berant, Andrew McCallum, and Sameer Singh (eds.), *3rd Conference on Automated Knowledge Base Construction, AKBC 2021, Virtual, October 4-8, 2021*, 2021. doi: 10.24432/C5F30Z. URL https://doi.org/10.24432/C5F30Z.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. TaPas: Weakly supervised table parsing via pre-training. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4320–4333, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.398. URL https://aclanthology.org/2020.acl-main.398/.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. Metagpt: Meta programming for A multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=VtmBAGCN7o.

Chenxu Hu, Jie Fu, Chenzhuang Du, Simian Luo, Junbo Zhao, and Hang Zhao. Chatdb: Augmenting llms with databases as their symbolic memory. *CoRR*, abs/2306.03901, 2023. doi: 10.48550/ARXIV.2306.03901. URL https://doi.org/10.48550/arXiv.2306.03901.

Xueyu Hu, Ziyu Zhao, Shuang Wei, Ziwei Chai, Qianli Ma, Guoyin Wang, Xuwu Wang, Jing Su, Jingjing Xu, Ming Zhu, Yao Cheng, Jianbo Yuan, Jiwei Li, Kun Kuang, Yang Yang, Hongxia Yang, and Fei Wu. Infiagent-dabench: Evaluating agents on data analysis tasks. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pp. 19544–19572. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/hu24s.html.

Yanwei Huang, Yunfan Zhou, Ran Chen, Changhao Pan, Xinhuan Shu, Di Weng, and Yingcai Wu. Interactive table synthesis with natural language. *IEEE Trans. Vis. Comput. Graph.*, 30(9):6130–6145, 2024a. doi: 10.1109/TVCG.2023.3329120. URL https://doi.org/10.1109/TVCG.2023.3329120.

Yiming Huang, Jianwen Luo, Yan Yu, Yitong Zhang, Fangyu Lei, Yifan Wei, Shizhu He, Lifu Huang, Xiao Liu, Jun Zhao, and Kang Liu. DA-code: Agent data science code generation benchmark for large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 13487–13521, Miami, Florida, USA, November 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.748. URL https://aclanthology.org/2024.emnlp-main.748/.

Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. Search-based neural structured learning for sequential question answering. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1821–1831, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1167. URL https://aclanthology.org/P17-1167/.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S. Yu. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Trans. Neural Networks Learn. Syst.*, 33(2):494–514, 2022. doi: 10.1109/TNNLS.2021.3070843. URL https://doi.org/10.1109/TNNLS.2021.3070843.

Yunjie Ji, Hao Liu, Bolei He, Xinyan Xiao, Hua Wu, and Yanhua Yu. Diversified multiple instance learning for document-level multi-aspect sentiment classification. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7012–7023, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.570. URL https://aclanthology.org/2020.emnlp-main.570/.

Zhengbao Jiang, Yi Mao, Pengcheng He, Graham Neubig, and Weizhu Chen. OmniTab: Pretraining with natural and synthetic data for few-shot table-based question answering. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 932–942, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.68. URL https://aclanthology.org/2022.naacl-main.68/.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? A large-scale open domain question answering dataset from medical exams. *CoRR*, abs/2009.13081, 2020. URL https://arxiv.org/abs/2009.13081.

Ruochun Jin, Xiyue Wang, Dong Wang, Haoqi Zheng, Yunpeng Qi, Silin Yang, and Meng Zhang. TALON: A multi-agent framework for long-table exploration and question answering. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 27397–27413, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1393. URL https://aclanthology.org/2025.emnlp-main.1393/.

George Katsogiannis-Meimarakis and Georgia Koutrika. A survey on deep learning approaches for text-to-sql. *VLDB J.*, 32(4):905–936, 2023. doi: 10.1007/S00778-022-00776-8. URL https://doi.org/10.1007/s00778-022-00776-8.

Subhendu Khatuya, Shashwat Naidu, Pawan Goyal, and Niloy Ganguly. Program of thoughts for financial reasoning: Leveraging dynamic in-context examples and generative retrieval. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 31006–31018, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1577. URL https://aclanthology.org/2025.emnlp-main.1577/.

Wei-Jen Ko, Avik Ray, Yilin Shen, and Hongxia Jin. Generating dialogue responses from a semantic latent space. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4339–4349, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.352. URL https://aclanthology.org/2020.emnlp-main.352/.

Fangyu Lei, Jixuan Chen, Yuxiao Ye, Ruisheng Cao, Dongchan Shin, Hongjin Su, Zhaoqing Suo, Hongcheng Gao, Wenjing Hu, Pengcheng Yin, Victor Zhong, Caiming Xiong, Ruoxi Sun, Qian Liu, Sida Wang, and Tao Yu. Spider 2.0: Evaluating language models on real-world enterprise text-to-sql workflows. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=XmProj9cPs.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Bei Li, Ziyang Wang, Hui Liu, Yufan Jiang, Quan Du, Tong Xiao, Huizhen Wang, and Jingbo Zhu. Shallow-to-deep training for neural machine translation. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 995–1005, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.72. URL https://aclanthology.org/2020.emnlp-main.72/.

Boyan Li, Jiayi Zhang, Ju Fan, Yanwei Xu, Chong Chen, Nan Tang, and Yuyu Luo. Alpha-sql: Zero-shot text-to-sql using monte carlo tree search. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/li25dt.html.

Hongxin Li, Jingran Su, Yuntao Chen, Qing Li, and Zhaoxiang Zhang. Sheetcopilot: Bringing software productivity to the next level through large language models. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023a. URL http://papers.nips.cc/paper_files/paper/2023/hash/0ff30c4bf31db0119a6219e0d250e037-Abstract-Conference.html.

Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Ruiying Geng, Nan Huo, Xuanhe Zhou, Chenhao Ma, Guoliang Li, Kevin Chen-Chuan Chang, Fei Huang, Reynold Cheng, and Yongbin Li. Can LLM already serve as A database interface? A big bench for large-scale database grounded text-to-sqls. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023b. URL http://papers.nips.cc/paper_files/paper/2023/hash/83fc8fab1710363050bbd1d4b8cc0021-Abstract-Datasets_and_Benchmarks.html.

Yongqi Li, Wenjie Li, and Liqiang Nie. MMCoQA: Conversational question answering over text, tables, and images. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4220–4231, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.290. URL https://aclanthology.org/2022.acl-long.290/.

Qi Liu, Zihuiwen Ye, Tao Yu, Linfeng Song, and Phil Blunsom. Augmenting multi-turn text-to-SQL datasets with self-play. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 5608–5620, Abu Dhabi, United Arab Emirates, December 2022a. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.411. URL https://aclanthology.org/2022.findings-emnlp.411/.

Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. TAPEX: table pre-training via learning a neural SQL executor. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022b. URL https://openreview.net/forum?id=O50443AsCP.

Xinyu Liu, Shuyu Shen, Boyan Li, Peixian Ma, Runzhi Jiang, Yuyu Luo, Yuxin Zhang, Ju Fan, Guoliang Li, and Nan Tang. A survey of NL2SQL with large language models: Where are we, and where are we going? *CoRR*, abs/2408.05109, 2024. doi: 10.48550/ARXIV.2408.05109. URL https://doi.org/10.48550/arXiv.2408.05109.

Zeyao Ma, Bohan Zhang, Jing Zhang, Jifan Yu, Xiaokang Zhang, Xiaohan Zhang, Sijia Luo, Xi Wang, and Jie Tang. Spreadsheetbench: Towards challenging real world spreadsheet manipulation. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/ac840df270ac537dd74530a15c332684-Abstract-Datasets_and_Benchmarks_Track.html.

Kai Nakamura, Sharon Levy, Yi-Lin Tuan, Wenhu Chen, and William Yang Wang. HybriDialogue: An information-seeking dialogue dataset grounded on tabular and textual data. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 481–492, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.41. URL https://aclanthology.org/2022.findings-acl.41/.

Jaehyun Nam, Jinsung Yoon, Jiefeng Chen, Raj Sinha, Jinwoo Shin, and Tomas Pfister. Ds-star: Data science agent for solving diverse tasks across heterogeneous formats and open-ended queries, 2025. URL https://arxiv.org/abs/2509.21825.

Linyong Nan, Chiachun Hsieh, Ziming Mao, Xi Victoria Lin, Neha Verma, Rui Zhang, Wojciech Kryściński, Hailey Schoelkopf, Riley Kong, Xiangru Tang, Mutethia Mutuma, Ben Rosand, Isabel Trindade, Renusree Bandaru, Jacob Cunningham, Caiming Xiong, and Dragomir Radev. FeTaQA: Free-form table question answering. *Transactions of the Association for Computational Linguistics*, 10:35–49, 2022. doi: 10.1162/tacl_a_00446. URL https://aclanthology.org/2022.tacl-1.3/.

Arpit Narechania, Arjun Srinivasan, and John T. Stasko. NL4DV: A toolkit for generating analytic specifications for data visualization from natural language queries. *IEEE Trans. Vis. Comput. Graph.*, 27(2):369–379, 2021. doi: 10.1109/TVCG.2020.3030378. URL https://doi.org/10.1109/TVCG.2020.3030378.

Matthew J. Page, Joanne E. McKenzie, Patrick M. Bossuyt, Isabelle Boutron, Tammy C. Hoffmann, Cynthia D. Mulrow, Larissa Shamseer, Jennifer M. Tetzlaff, Elie A. Akl, Sue E. Brennan, Roger Chou, Julie Glanville, Jeremy M. Grimshaw, Asbjørn Hróbjartsson, Manoj M. Lalu, Tianjing Li, Elizabeth W. Loder, Evan Mayo-Wilson, Steve McDonald, Luke A. McGuinness, Lesley A. Stewart, James Thomas, Andrea C. Tricco, Vivian A. Welch, Penny Whiting, and David Moher. The prisma 2020 statement: an updated guideline for reporting systematic reviews. *Systematic Reviews*, 10, 2021. doi: 10.1186/s13643-021-01626-4.

Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. In Chengqing Zong and Michael Strube (eds.), *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1470–1480, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1142. URL https://aclanthology.org/P15-1142/.

Mohammadreza Pourreza and Davood Rafiei. DIN-SQL: decomposed in-context learning of text-to-sql with self-correction. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/72223cc66f63ca1aa59edaec1b3670e6-Abstract-Conference.html.

Mohammadreza Pourreza, Hailong Li, Ruoxi Sun, Yeounoh Chung, Shayan Talaei, Gaurav Tarlok Kakkar, Yu Gan, Amin Saberi, Fatma Ozcan, and Sercan Ö. Arik. CHASE-SQL: multi-path reasoning and preference optimized candidate selection in text-to-sql. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=CvGqMD5OtX.

Siva Reddy, Danqi Chen, and Christopher D. Manning. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, 2019. doi: 10.1162/tacl_a_00266. URL https://aclanthology.org/Q19-1016/.

Aécio Santos, Eduardo H. M. Pena, Roque Lopez, and Juliana Freire. Interactive data harmonization with llm agents: Opportunities and challenges, 2025. URL https://arxiv.org/abs/2502.07132.

Christopher Scaffidi, Mary Shaw, and Brad A. Myers. Estimating the numbers of end users and end user programmers. In *2005 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2005), 21-24 September 2005, Dallas, TX, USA*, pp. 207–214. IEEE Computer Society, 2005. doi: 10.1109/VLHCC.2005.34. URL https://doi.org/10.1109/VLHCC.2005.34.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html.

Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. PICARD: Parsing incrementally for constrained auto-regressive decoding from language models. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 9895–9901, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.779. URL https://aclanthology.org/2021.emnlp-main.779/.

Vidya Setlur, Sarah E. Battersby, Melanie Tory, Rich Gossweiler, and Angel X. Chang. Eviza: A natural language interface for visual analysis. In Jun Rekimoto, Takeo Igarashi, Jacob O. Wobbrock, and Daniel Avrahami (eds.), *Proceedings of the 29th Annual Symposium on User Interface Software and Technology, UIST 2016, Tokyo, Japan, October 16-19, 2016*, pp. 365–377. ACM, 2016. doi: 10.1145/2984511.2984588. URL https://doi.org/10.1145/2984511.2984588.

Wenqi Shi, Ran Xu, Yuchen Zhuang, Yue Yu, Jieyu Zhang, Hang Wu, Yuanda Zhu, Joyce C. Ho, Carl Yang, and May Dongmei Wang. EHRAgent: Code empowers large language models for few-shot complex tabular reasoning on electronic health records. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 22315–22339, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1245. URL https://aclanthology.org/2024.emnlp-main.1245/.

Arjun Srinivasan and John T. Stasko. Orko: Facilitating multimodal interaction for visual exploration and analysis of networks. *IEEE Trans. Vis. Comput. Graph.*, 24(1):511–521, 2018. doi: 10.1109/TVCG.2017.2745219. URL https://doi.org/10.1109/TVCG.2017.2745219.

Aofeng Su, Aowen Wang, Chao Ye, Chen Zhou, Ga Zhang, Gang Chen, Guangcheng Zhu, Haobo Wang, Haokai Xu, Hao Chen, Haoze Li, Haoxuan Lan, Jiaming Tian, Jing Yuan, Junbo Zhao, Junlin Zhou, Kaizhe Shou, Liangyu Zha, Lin Long, Liyao Li, Pengzuo Wu, Qi Zhang, Qingyi Huang, Saisai Yang, Tao Zhang, Wentao Ye, Wufang Zhu, Xiaomeng Hu, Xijun Gu, Xinjie Sun, Xiang Li, Yuhang Yang, and Zhiqing Xiao. Tablegpt2: A large multimodal model with tabular data integration. *CoRR*, abs/2411.02059, 2024. doi: 10.48550/ARXIV.2411.02059. URL https://doi.org/10.48550/arXiv.2411.02059.

Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. Table meets LLM: can large language models understand structured table data? A benchmark and empirical study. In Luz Angelica Caudillo-Mata, Silvio Lattanzi, Andrés Muñoz Medina, Leman Akoglu, Aristides Gionis, and Sergei Vassilvitskii (eds.), *Proceedings of the 17th ACM International Conference on Web Search and Data Mining, WSDM 2024, Merida, Mexico, March 4-8, 2024*, pp. 645–654. ACM, 2024a. doi: 10.1145/3616855.3635752. URL https://doi.org/10.1145/3616855.3635752.

Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. Table meets LLM: can large language models understand structured table data? A benchmark and empirical study. In Luz Angelica Caudillo-Mata, Silvio Lattanzi, Andrés Muñoz Medina, Leman Akoglu, Aristides Gionis, and Sergei Vassilvitskii (eds.), *Proceedings of the 17th ACM International Conference on Web Search and Data Mining, WSDM 2024, Merida, Mexico, March 4-8, 2024*, pp. 645–654. ACM, 2024b. doi: 10.1145/3616855.3635752. URL https://doi.org/10.1145/3616855.3635752.

Anirudh Sundar, Jin Xu, William Gay, Christopher Richardson, and Larry Heck. cpapers: A dataset of situated and multimodal interactive conversations in scientific papers. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and

Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/7a19a9d527ed544d1272f07b0f8f934e-Abstract-Datasets_and_Benchmarks_Track.html.

Anirudh Sundar, Christopher Richardson, Adar Avsian, and Larry Heck. iTBLS: A dataset of interactive conversations over tabular information. In Shuaichen Chang, Madelon Hulsebos, Qian Liu, Wenhu Chen, and Huan Sun (eds.), *Proceedings of the 4th Table Representation Learning Workshop*, pp. 56–70, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-268-8. doi: 10.18653/v1/2025.trl-1.6. URL https://aclanthology.org/2025.trl-1.6/.

Anirudh S. Sundar and Larry Heck. cTBLS: Augmenting large language models with conversational tables. In Yun-Nung Chen and Abhinav Rastogi (eds.), *Proceedings of the 5th Workshop on NLP for Conversational AI (NLP4ConvAI 2023)*, pp. 59–70, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.nlp4convai-1.6. URL https://aclanthology.org/2023.nlp4convai-1.6/.

Jiaming Tian, Liyao Li, Wentao Ye, Haobo Wang, Lingxin Wang, Lihua Yu, Zujie Ren, Gang Chen, and Junbo Zhao. Toward real-world table agents: capabilities, workflows, and design principles for llm-based table intelligence. *World Wide Web (WWW)*, 29(2):16, 2026. doi: 10.1007/S11280-025-01399-Z. URL https://doi.org/10.1007/s11280-025-01399-z.

Yuzhang Tian, Jianbo Zhao, Haoyu Dong, Junyu Xiong, Shiyu Xia, Mengyu Zhou, Yun Lin, José Cambronero, Yeye He, Shi Han, and Dongmei Zhang. Spreadsheetllm: Encoding spreadsheets for large language models. *CoRR*, abs/2407.09025, 2024. doi: 10.48550/ARXIV.2407.09025. URL https://doi.org/10.48550/arXiv.2407.09025.

Svitlana Vakulenko, Shayne Longpre, Zhucheng Tu, and Raviteja Anantha. Question rewriting for conversational question answering. In Liane Lewin-Eytan, David Carmel, Elad Yom-Tov, Eugene Agichtein, and Evgeniy Gabrilovich (eds.), *WSDM '21, The Fourteenth ACM International Conference on Web Search and Data Mining, Virtual Event, Israel, March 8-12, 2021*, pp. 355–363. ACM, 2021. doi: 10.1145/3437963.3441748. URL https://doi.org/10.1145/3437963.3441748.

Bailin Wang, Richard Shin, Xiaodong Liu, Oleksandr Polozov, and Matthew Richardson. RAT-SQL: Relation-aware schema encoding and linking for text-to-SQL parsers. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7567–7578, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.677. URL https://aclanthology.org/2020.acl-main.677/.

Bing Wang, Changyu Ren, Jian Yang, Xinnian Liang, Jiaqi Bai, LinZheng Chai, Zhao Yan, Qian-Wen Zhang, Di Yin, Xing Sun, and Zhoujun Li. MAC-SQL: A multi-agent collaborative framework for text-to-SQL. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 540–557, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.36/.

Runze Wang, Zhen-Hua Ling, Jingbo Zhou, and Yu Hu. Tracking interaction states for multi-turn text-to-sql semantic parsing. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 13979–13987. AAAI Press, 2021. doi: 10.1609/AAAI.V35I16.17646. URL https://doi.org/10.1609/aaai.v35i16.17646.

Yuan Wang, Xuyang Wu, Hsin-Tai Wu, Zhiqiang Tao, and Yi Fang. Do large language models rank fairly? an empirical study on the fairness of LLMs as rankers. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5712–5724, Mexico City, Mexico, June 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.319. URL https://aclanthology.org/2024.naacl-long.319/.

Zilong Wang, Hao Zhang, Chun-Liang Li, Julian Martin Eisenschlos, Vincent Perot, Zifeng Wang, Lesly Miculicich, Yasuhisa Fujii, Jingbo Shang, Chen-Yu Lee, and Tomas Pfister. Chain-of-table: Evolving tables in the reasoning chain for table understanding. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024b. URL https://openreview.net/forum?id=4L0xnS4GQM.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

William A. Woods. Progress in natural language understanding: an application to lunar geology. In *American Federation of Information Processing Societies: 1973 National Computer Conference, 4-8 June 1973, New York, NY, USA*, AFIPS Conference Proceedings, pp. 441–450. AFIPS Press/ACM, 1973. doi: 10.1145/1499586.1499695. URL https://doi.org/10.1145/1499586.1499695.

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I. Wang, Victor Zhong, Bailin Wang, Chengzu Li, Connor Boyle, Ansong Ni, Ziyu Yao, Dragomir Radev, Caiming Xiong, Lingpeng Kong, Rui Zhang, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. UnifiedSKG: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 602–631, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.39. URL https://aclanthology.org/2022.emnlp-main.39/.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 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, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. TaBERT: Pretraining for joint understanding of textual and tabular data. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8413–8426, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.745. URL https://aclanthology.org/2020.acl-main.745/.

Peiying Yu, Guoxin Chen, and Jingjing Wang. Table-critic: A multi-agent framework for collaborative criticism and refinement in table reasoning. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 17432–17451, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.853. URL https://aclanthology.org/2025.acl-long.853/.

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. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3911–3921, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1425. URL https://aclanthology.org/D18-1425/.

Tao Yu, Rui Zhang, Heyang Er, Suyi Li, Eric Xue, Bo Pang, Xi Victoria Lin, Yi Chern Tan, Tianze Shi, Zihan Li, Youxuan Jiang, Michihiro Yasunaga, Sungrok Shim, Tao Chen, Alexander Fabbri, Zifan Li, Luyao Chen, Yuwen Zhang, Shreya Dixit, Vincent Zhang, Caiming Xiong, Richard Socher, Walter Lasecki, and Dragomir Radev. CoSQL: A conversational text-to-SQL challenge towards cross-domain natural language interfaces to databases. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.),

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 1962–1979, Hong Kong, China, November 2019a. Association for Computational Linguistics. doi: 10.18653/v1/D19-1204. URL https://aclanthology.org/D19-1204/.

Tao Yu, Rui Zhang, Michihiro Yasunaga, Yi Chern Tan, Xi Victoria Lin, Suyi Li, Heyang Er, Irene Li, Bo Pang, Tao Chen, Emily Ji, Shreya Dixit, David Proctor, Sungrok Shim, Jonathan Kraft, Vincent Zhang, Caiming Xiong, Richard Socher, and Dragomir Radev. SParC: Cross-domain semantic parsing in context. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4511–4523, Florence, Italy, July 2019b. Association for Computational Linguistics. doi: 10.18653/v1/P19-1443. URL https://aclanthology.org/P19-1443/.

Munazza Zaib, Wei Emma Zhang, Quan Z. Sheng, Adnan Mahmood, and Yang Zhang. Conversational question answering: A survey, 2021. URL https://arxiv.org/abs/2106.00874.

Hanchong Zhang, Ruisheng Cao, Lu Chen, Hongshen Xu, and Kai Yu. ACT-SQL: In-context learning for text-to-SQL with automatically-generated chain-of-thought. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 3501–3532, Singapore, December 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.227. URL https://aclanthology.org/2023.findings-emnlp.227/.

Rui Zhang, Tao Yu, Heyang Er, Sungrok Shim, Eric Xue, Xi Victoria Lin, Tianze Shi, Caiming Xiong, Richard Socher, and Dragomir Radev. Editing-based SQL query generation for cross-domain context-dependent questions. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5338–5349, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1537. URL https://aclanthology.org/D19-1537/.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=SkeHuCVFDr.

Weixu Zhang, Yifei Wang, Yuanfeng Song, Victor Junqiu Wei, Yuxing Tian, Yiyan Qi, Jonathan H. Chan, Raymond Chi-Wing Wong, and Haiqin Yang. Natural language interfaces for tabular data querying and visualization: A survey. *IEEE Trans. Knowl. Data Eng.*, 36(11):6699–6718, 2024. doi: 10.1109/TKDE.2024.3400824. URL https://doi.org/10.1109/TKDE.2024.3400824.

Wenqi Zhang, Yongliang Shen, Weiming Lu, and Yueting Zhuang. Data-copilot: Bridging billions of data and humans with autonomous workflow. *CoRR*, abs/2306.07209, 2023b. doi: 10.48550/ARXIV.2306.07209. URL https://doi.org/10.48550/arXiv.2306.07209.

Xiaokang Zhang, Sijia Luo, Bohan Zhang, Zeyao Ma, Jing Zhang, Yang Li, Guanlin Li, Zijun Yao, Kangli Xu, Jinchang Zhou, Daniel Zhang-Li, Jifan Yu, Shu Zhao, Juanzi Li, and Jie Tang. TableLLM: Enabling tabular data manipulation by LLMs in real office usage scenarios. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 10315–10344, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.538. URL https://aclanthology.org/2025.findings-acl.538/.

Weichao Zhao, Hao Feng, Qi Liu, Jingqun Tang, Binghong Wu, Lei Liao, Shu Wei, Yongjie Ye, Hao Liu, Wengang Zhou, Houqiang Li, and Can Huang. Tabpedia: Towards comprehensive visual table understanding with concept synergy. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/0d97fe65d7a1dc12a05642d9fa4cd578-Abstract-Conference.html.

Yilun Zhao, Yunxiang Li, Chenying Li, and Rui Zhang. MultiHiertt: Numerical reasoning over multi hierarchical tabular and textual data. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6588–6600, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.454. URL https://aclanthology.org/2022.acl-long.454/.

Yanzhao Zheng, Haibin Wang, Baohua Dong, Xingjun Wang, and Changshan Li. HIE-SQL: History information enhanced network for context-dependent text-to-SQL semantic parsing. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2997–3007, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.236. URL https://aclanthology.org/2022.findings-acl.236/.

Shanshan Zhong, Jinghui Qin, Zhongzhan Huang, and Daifeng Li. CEM: Machine-human chatting handoff via causal-enhance module. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 3242–3253, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.213. URL https://aclanthology.org/2022.emnlp-main.213/.

Zexuan Zhong, Dan Friedman, and Danqi Chen. Factual probing is [MASK]: Learning vs. learning to recall. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5017–5033, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.398. URL https://aclanthology.org/2021.naacl-main.398/.

Xuanhe Zhou, Guoliang Li, Zhaoyan Sun, Zhiyuan Liu, Weize Chen, Jianming Wu, Jiesi Liu, Ruohang Feng, and Guoyang Zeng. D-bot: Database diagnosis system using large language models. *Proc. VLDB Endow.*, 17(10):2514–2527, 2024. doi: 10.14778/3675034.3675043. URL https://www.vldb.org/pvldb/vol17/p2514-li.pdf.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *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)*, pp. 3277–3287, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.254. URL https://aclanthology.org/2021.acl-long.254/.

Jun-Peng Zhu, Peng Cai, Kai Xu, Li Li, Yishen Sun, Shuai Zhou, Haihuang Su, Liu Tang, and Qi Liu. Autotqa: Towards autonomous tabular question answering through multi-agent large language models. *Proc. VLDB Endow.*, 17(12):3920–3933, 2024. doi: 10.14778/3685800.3685816. URL https://www.vldb.org/pvldb/vol17/p3920-zhu.pdf.

Ruiyan Zhu, Xi Cheng, Ke Liu, Brian Zhu, Daniel Jin, Neeraj Parihar, Zhoutian Xu, and Oliver Gao. Sheetmind: An end-to-end llm-powered multi-agent framework for spreadsheet automation. *CoRR*, abs/2506.12339, 2025. doi: 10.48550/ARXIV.2506.12339. URL https://doi.org/10.48550/arXiv.2506.12339.

A Appendix

You may include other additional sections here.