

ConversaDB: A Context-Aware Query Engine for Conversational Database Interactions

Emily R. Nakamura
University of Washington
Paul G. Allen School of Computer
Science
Seattle, WA, USA
enakamura@cs.washington.edu

Carlos A. Mendez
ETH Zurich
Department of Computer Science
Zurich, Switzerland
mendez@inf.ethz.ch

Priya S. Gupta
Microsoft Research
Redmond, WA, USA
priyag@microsoft.com

ABSTRACT

Natural language interfaces to databases (NLDBs) have long been a goal of the database and NLP communities. While recent advances in large language models (LLMs) have dramatically improved text-to-SQL accuracy, existing systems treat each query independently, ignoring the conversational context that is natural in real-world database interactions. We present ConversaDB, a context-aware query engine that maintains a multi-turn dialogue state to resolve ambiguities, handle co-references, and support follow-up queries. Our system introduces three key innovations: (1) a dialogue state tracker that maintains a structured representation of the conversation history, including referenced tables, columns, and filter conditions; (2) a context-aware SQL generation module that conditions on both the current utterance and the dialogue state; and (3) an interactive disambiguation protocol that proactively seeks clarification when query intent is uncertain. We evaluate ConversaDB on the SPaC and CoSQL benchmarks, achieving 71.4% and 53.8% question match accuracy respectively, representing improvements of 6.2% and 8.1% over the previous state of the art. A user study with 48 participants confirms that ConversaDB significantly reduces the number of interaction turns required to complete complex analytical tasks.

KEYWORDS

natural language interfaces, text-to-SQL, conversational AI, database systems, dialogue state tracking

1 INTRODUCTION

Databases store vast amounts of structured information, yet accessing this data requires proficiency in query languages such as SQL. Natural language interfaces to databases (NLDBs) aim to bridge this gap by allowing users to express their information needs in plain language. Recent progress in large language models has led to dramatic improvements in single-turn text-to-SQL systems, with state-of-the-art models achieving over 80% accuracy on the Spider benchmark [1].

However, real-world database interactions are inherently conversational. Users rarely formulate their complete information need in a single query. Instead, they engage in multi-turn dialogues, refining their queries, asking follow-up questions, and exploring the data incrementally. Consider the following example interaction:

- (1) “Show me sales by region for Q4 2024.”
- (2) “What about the previous quarter?”
- (3) “Sort those by revenue.”
- (4) “Which region had the highest growth?”

Each utterance after the first requires understanding the conversational context to generate the correct SQL query. Utterance 2 requires resolving “the previous quarter” relative to Q4 2024. Utterance 3 refers to “those,” which co-references the results from utterance 2. These phenomena—temporal references, co-references, and ellipsis—are pervasive in natural dialogue but poorly handled by single-turn systems.

Contributions. We make the following contributions:

- We introduce ConversaDB, a context-aware query engine that maintains structured dialogue state for multi-turn database interactions.
- We propose a novel context-aware SQL generation architecture that combines dialogue state tracking with constrained decoding.
- We design an interactive disambiguation protocol that reduces errors by proactively seeking clarification.
- We achieve new state-of-the-art results on SPaC and CoSQL benchmarks and validate our system through a comprehensive user study.

2 RELATED WORK

2.1 Text-to-SQL

The text-to-SQL task has seen rapid progress in recent years. Early approaches used rule-based parsing and template filling [2]. Neural approaches, beginning with Seq2SQL [3], framed the problem as sequence-to-sequence translation. More recent work leverages pre-trained language models, with systems like RESDSQL [4] achieving over 80% exact match accuracy on Spider.

2.2 Conversational Text-to-SQL

SPaC [5] and CoSQL [6] introduced benchmarks for context-dependent text-to-SQL. EditSQL [7] proposed editing the previous SQL query to handle follow-up questions. IGSQ [8] used an interaction graph to capture cross-turn dependencies. Despite these advances, handling complex co-references and temporal expressions remains challenging.

3 SYSTEM ARCHITECTURE

ConversaDB consists of three main components: (1) a Dialogue State Tracker (DST), (2) a Context-Aware SQL Generator (CASG), and (3) an Interactive Disambiguation Module (IDM). Figure 1 illustrates the overall architecture.

Table 1: Question match accuracy (%) on SPaRC and CoSQL development sets.

Model	SPaRC	CoSQL
EditSQL (2019)	47.2	31.4
IGSQL (2020)	50.7	36.8
HIE-SQL (2022)	60.1	42.3
STAR (2023)	65.2	45.7
ConversaDB	71.4	53.8

3.1 Dialogue State Tracker

The DST maintains a structured representation S_t at each turn t that captures the essential context from the conversation history:

$$S_t = (\mathcal{T}_t, C_t, \mathcal{F}_t, \mathcal{R}_t, Q_{t-1}) \quad (1)$$

where $\mathcal{T}_t$ is the set of referenced tables, C_t is the set of referenced columns, $\mathcal{F}_t$ represents active filter conditions, $\mathcal{R}_t$ captures the result schema from the previous query, and Q_{t-1} is the most recent SQL query. At each turn, the DST updates the state based on the new utterance u_t :

$$S_t = \text{DST}(S_{t-1}, u_t, \mathcal{DB}) \quad (2)$$

where $\mathcal{DB}$ denotes the database schema.

3.2 Context-Aware SQL Generator

The CASG generates SQL queries conditioned on both the current utterance and the dialogue state. We encode the input as a linearized sequence:

$$x_t = [u_t; \text{serialize}(S_t); \text{serialize}(\mathcal{DB})] \quad (3)$$

The SQL query is generated autoregressively with schema-aware constrained decoding that ensures syntactic validity. We fine-tune a pre-trained T5-large model with the following objective:

$$\mathcal{L} = - \sum_{t=1}^T \sum_{j=1}^{|q_t|} \log P(q_t^j \mid q_t^{<j}, x_t; \theta) \quad (4)$$

3.3 Interactive Disambiguation

When the model’s confidence in the generated SQL falls below threshold τ , the IDM generates a clarification question. We compute confidence as the geometric mean of token-level probabilities:

$$\text{conf}(q_t) = \left(\prod_{j=1}^{|q_t|} P(q_t^j \mid q_t^{<j}, x_t) \right)^{1/|q_t|} \quad (5)$$

4 EVALUATION

4.1 Benchmark Results

Table 1 shows results on SPaRC and CoSQL.

Table 2: Ablation study on SPaRC development set.

Configuration	QM Acc. (%)
Full ConversaDB	71.4
– Dialogue State Tracker	63.8
– Constrained Decoding	67.1
– Interactive Disambiguation	69.2
– Pre-training	58.4

4.2 User Study

We conducted a user study with 48 participants (24 SQL-proficient, 24 non-technical) who completed 8 analytical tasks using ConversaDB and a single-turn baseline. ConversaDB reduced the average number of interaction turns from 6.3 to 3.8 and increased task completion rate from 72% to 91%. Participants rated ConversaDB 4.2/5.0 on naturalness compared to 2.8/5.0 for the baseline.

4.3 Ablation Study

Table 2 shows the contribution of each component.

5 CONCLUSION

We presented ConversaDB, a context-aware query engine that brings conversational capabilities to natural language database interfaces. By maintaining structured dialogue state and employing context-aware SQL generation with interactive disambiguation, ConversaDB achieves state-of-the-art results on multi-turn text-to-SQL benchmarks. Our user study confirms that the conversational paradigm significantly improves the database interaction experience for both technical and non-technical users.

ACKNOWLEDGMENTS

This research was supported by NSF Award IIS-2238811 and a gift from Microsoft Research. We thank the anonymous reviewers for their constructive feedback.

REFERENCES

- [1] T. Yu et al., “Spider: A large-scale human-labeled dataset for complex and cross-database semantic parsing and text-to-SQL task,” in *EMNLP*, 2018, pp. 3911–3921.
- [2] F. Li and H. V. Jagadish, “Constructing an interactive natural language interface for relational databases,” *PVLDB*, vol. 8, no. 1, pp. 73–84, 2014.
- [3] V. Zhong, C. Xiong, and R. Socher, “Seq2SQL: Generating structured queries from natural language using reinforcement learning,” *arXiv:1709.00103*, 2017.
- [4] H. Li et al., “RESDSL: Decoupling schema linking and skeleton parsing for text-to-SQL,” in *AAAI*, 2023.
- [5] T. Yu et al., “SPaRC: Cross-domain semantic parsing in context,” in *ACL*, 2019, pp. 4511–4523.
- [6] T. Yu et al., “CoSQL: A conversational text-to-SQL challenge towards cross-domain natural language interfaces to databases,” in *EMNLP*, 2019.
- [7] R. Zhang et al., “Editing-based SQL query generation for cross-domain context-dependent questions,” in *EMNLP*, 2019.
- [8] Y. Cai and B. Wan, “IGSQL: Database schema interaction graph based neural model for context-dependent text-to-SQL generation,” in *EMNLP*, 2020.