Transparent SQL-of-Thought: Explainable Multi-Agent Text-to-SQL Generation

Advait Shinde
Toney Zhen
Jasper Morgal

Advait Shinde
Professional Vibe Coder and Photographer

Toney Zhen
Professional Spectator, Cheerleader, Benchwarmer

Jasper Paul Morgal
Also Professional Vibe Coder

Photos courtesy of Advait Shinde
Taken during company bonding offsite hike

AGENDA

The Challenge

Convert natural language queries into SQL queries

Traditional Approaches

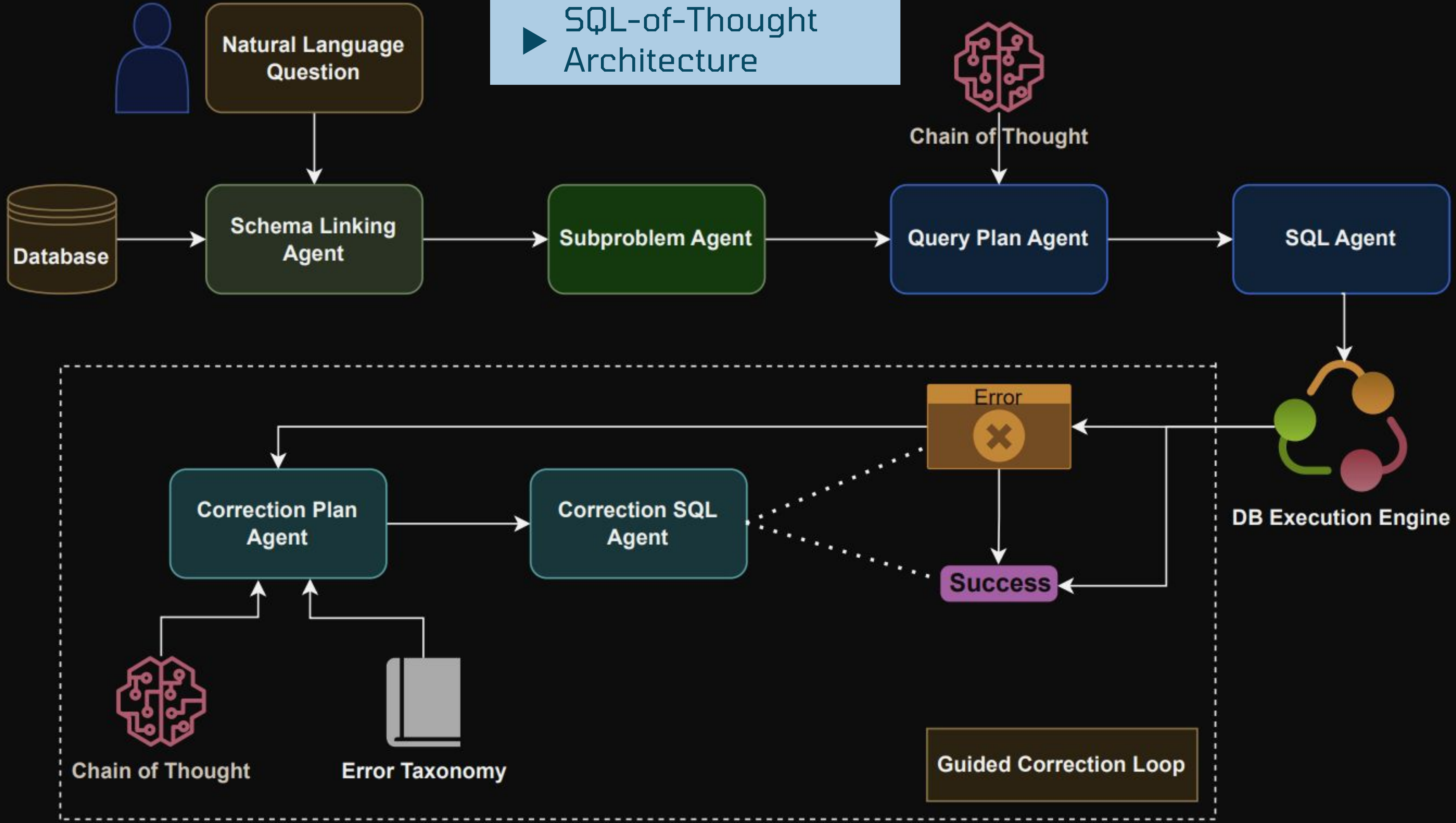
- ❌ Single-shot generation
- ❌ No structured reasoning
- ❌ Generic error feedback
- ❌ Blind refinement

Many generated queries are syntactically valid but logically incorrect. Execution errors alone don’t provide sufficient guidance for query correction.

Real World Needs

- ✅ Complex cross-domain queries
- ✅ Multi-table joins
- ✅ Logical error detection
- ✅ Iterative refinement

SQL-of-Thought Architecture



GUIDED ERROR CORRECTION : 9 Categories, 31 Error Types

- **Syntax**: sql_syntax_error, invalid_alias
- **Sub Query**: unused, missing, correlation_error
- **Set Operations**: union/intersect/except missing
- **Value Errors**: hardcoded_value, format_wrong
- **Join Errors**: join_missing, wrong_type, extra_table
- **Aggregation**: agg_no_groupby, having_vs_where
- **Schema Link**: table/col_missing, ambiguous_col, incorrect_fk
- **Filter Errors**: where_missing, wrong_col, type_mismatch
- **Others**: order_by/limit missing, extra_values

TIMELINE

🔗 GitHub: https://github.com/Jasper-256/sql_of_thought_recreation

📄 Paper: arXiv:2509.00581

We are here

01 Implemented paper from scratch

02 Replicated paper's SOTA results

03 Extended paper by adding schemas to every step

04 Ran benchmark tests with modern models

05 Added pipeline visualizer to understand the agent's reasoning

Our Implementation
Production Ready SQL-of-Thought Framework

Core Implementation

- Full 6-agent pipeline (LangGraph)
- Error taxonomy with 31 subtypes
- Chain-of-thought reasoning
- Guided correction loop
- Support for multiple LLM provider models

Benchmarking

- Spider, Spider-Realistic, Spider-SYN
- Automatic dataset download
- SQLite execution & validation
- CSV output with all metrics

Tech Stack

- Langgraph
- Langchain
- Open AI API
- Hugging Face Datasets

Technical Architecture: Deep Dive

1. LangGraph State Management

1. TypedDict State Schema
2. Shared State
3. Conditional Edges

   E.g. (`execute → (needs_correction && attempt < max) ? correction_plan : END`)

2. Schema Introspection

1. SQLite PRAGMA queries used to extract database structure
2. Raw DB metadata into LLM-friendly format

3. Validation

1. Syntax: SQL parsing via sqlite3

2. Execution: Catch runtime exceptions

3. Semantic: Compare with gold results

4. Error Detection Strategy

Two Correction Triggers:

A. Exception-based (syntax/runtime)

B. Result mismatch (logical errors)

Taxonomy Injection:

→ JSON taxonomy loaded at init

5. Models used

1. GPT-5.1 (no thinking)
2. GPT-5-mini (high)
3. Claude Opus 4.5

4. Design Patterns

1. Modularity
2. Type Safety
3. Error Resilience
4. Observability

OUR IMPROVEMENTS

1. Real-Time Visualization UI

🎨 Live NiceGUI-based dashboard

📊 View all agent inputs/outputs in real-time

🔍 Inspect system & user prompts for each step

📈 Track benchmark progress & metrics live

2. Advanced Model Support

🧠 Reasoning model integration (claude opus 4.5 and gpt-5-mini)

⚡ Configurable reasoning effort (low/medium/high)

🔄 Easy model switching via CLI

🚀 Measurable performance gains with modern LLMs

3. Enhanced Debugging

🔗 Pipeline visualizer for understanding agent reasoning

🛠️ Visual representation of reasoning chain

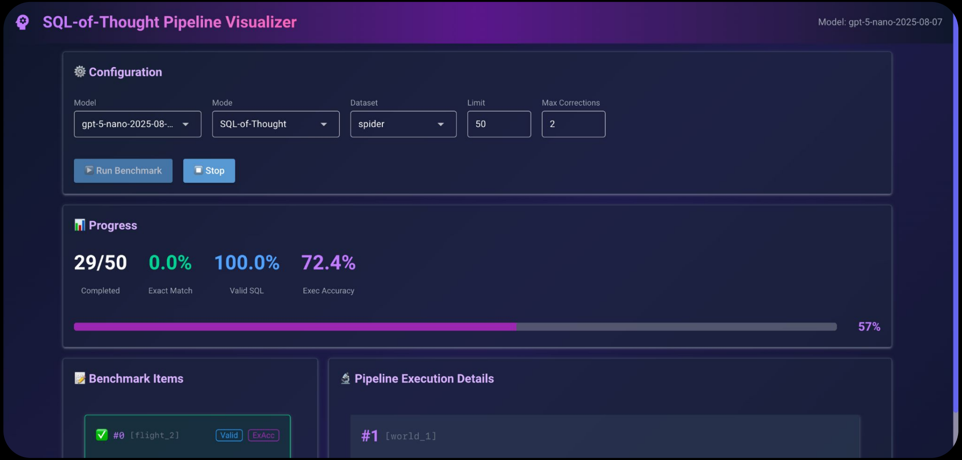
🐛 Improved debugging abilities for developers

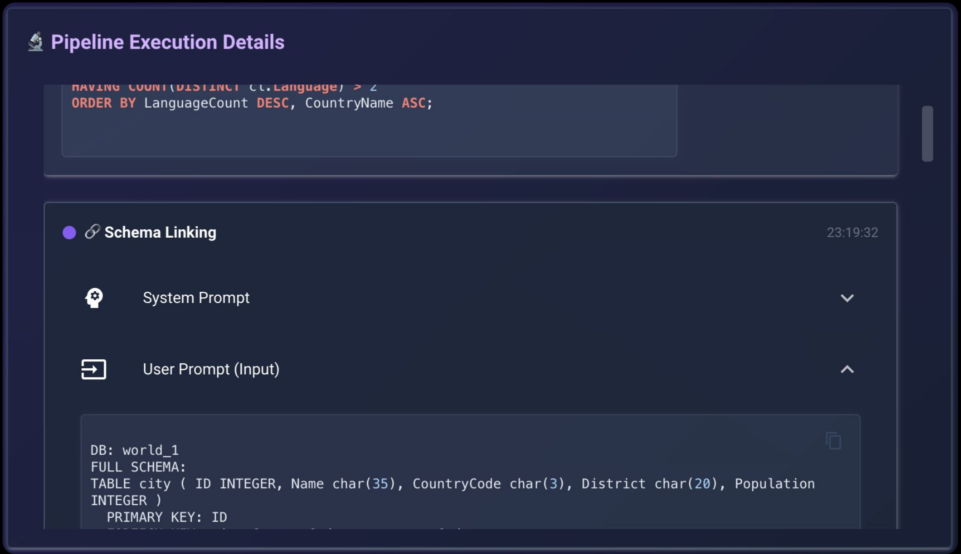
4. Architecture Improvements

📋 Provided the full schema to the SQL agent so that it has all relevant context

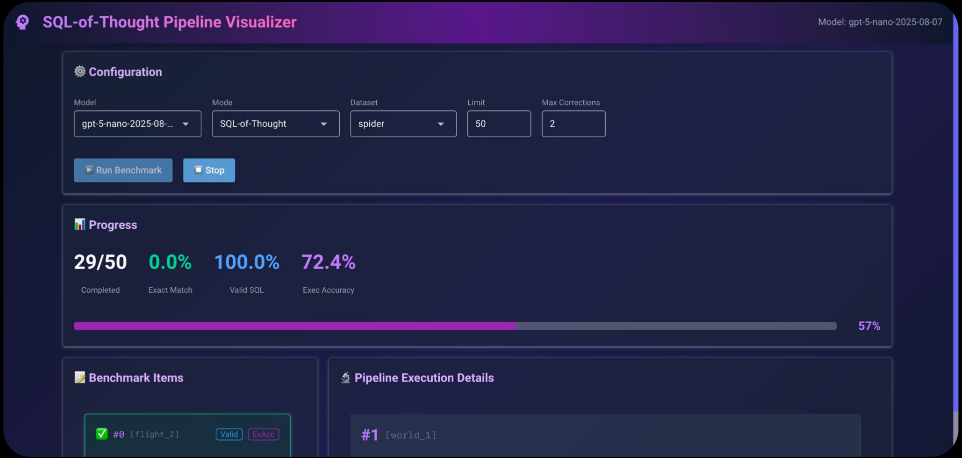
🏗️ Optimized architecture for better results

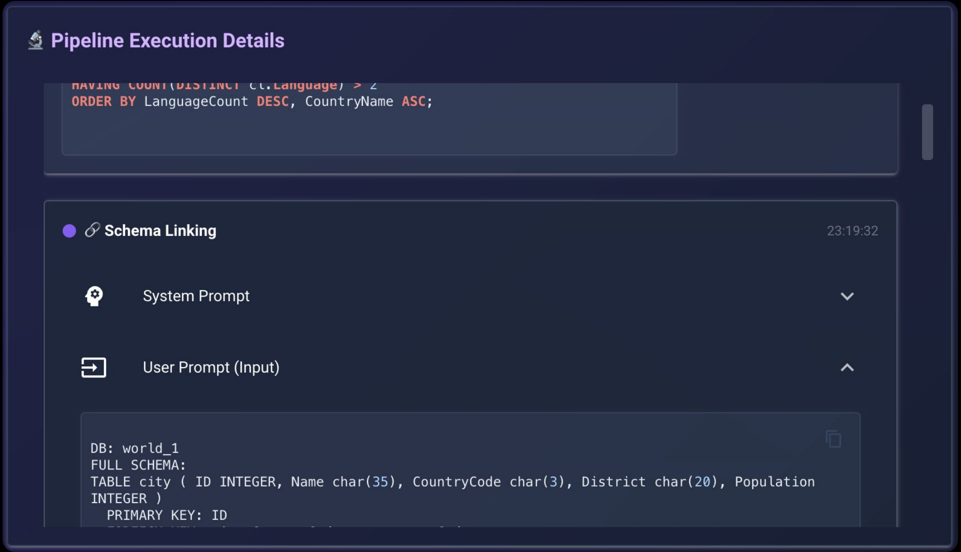
Pipeline Visualizer





Pipeline Visualizer





RESULTS

Metrics

92%

Highest Achieved

Execution Accuracy

Results Match

28%

Highest Achieved

Exact Match

SQL Query String Equality

100%

Highest Achieved

Valid SQL

Parses and Executes

RESULTS

SQL-of-Thought paper's execution accuracy on Spider benchmark:

Model	SQL-of-Thought
GPT-4	72.3%
Claude Opus 3	91.59%

Our execution accuracy on Spider benchmark:

Model	Base LLM	SQL-of-Thought
GPT-5.1 (no thinking)	36/50 (72%)	36/50 (72%)
GPT-5-mini (high)	41/50 (82%)	40/50 (80%)
Claude Opus 4.5	43/50 (86%)	46/50 (92%)

(Measured on 50 random queries from Spider)

THANK YOU

Any questions?