

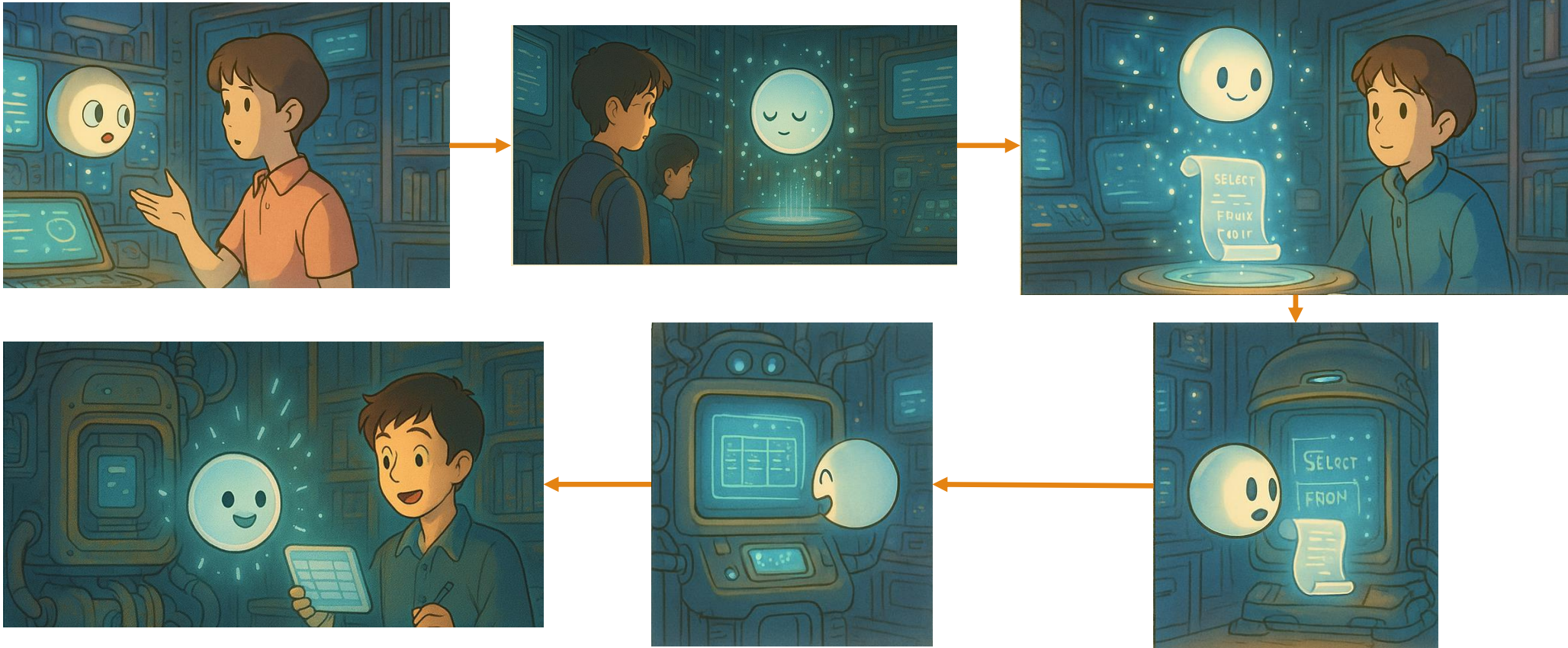
Optimizing Text-to-SQL Generation: A Dynamic Feedback Approach

-Vamsi Saripudi

04-09-2025

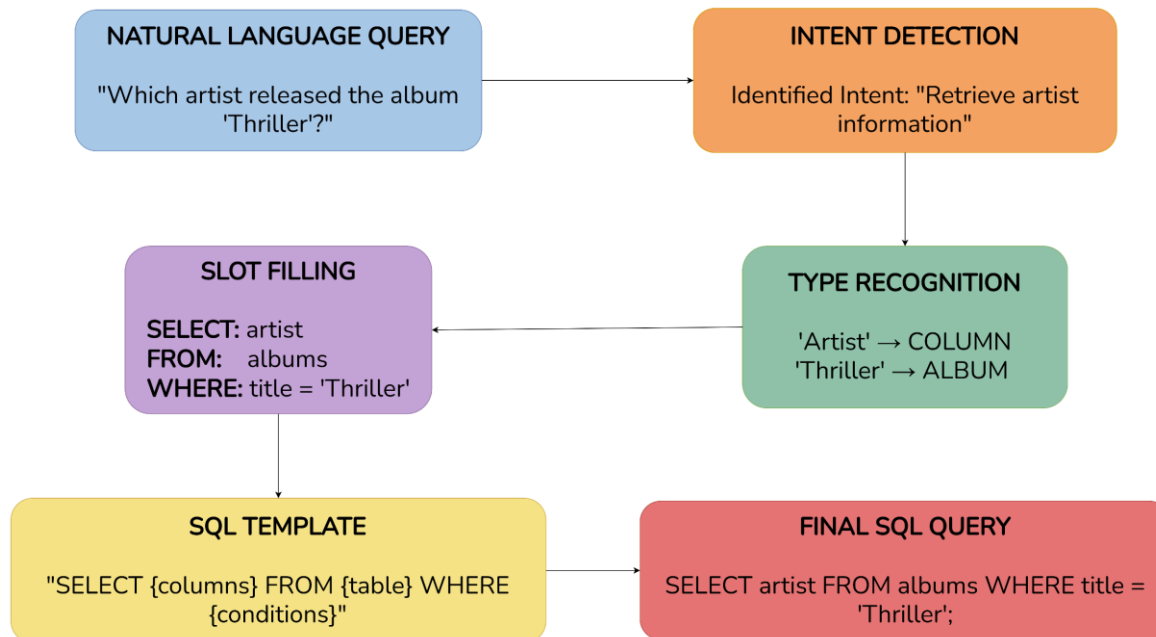


Introduction

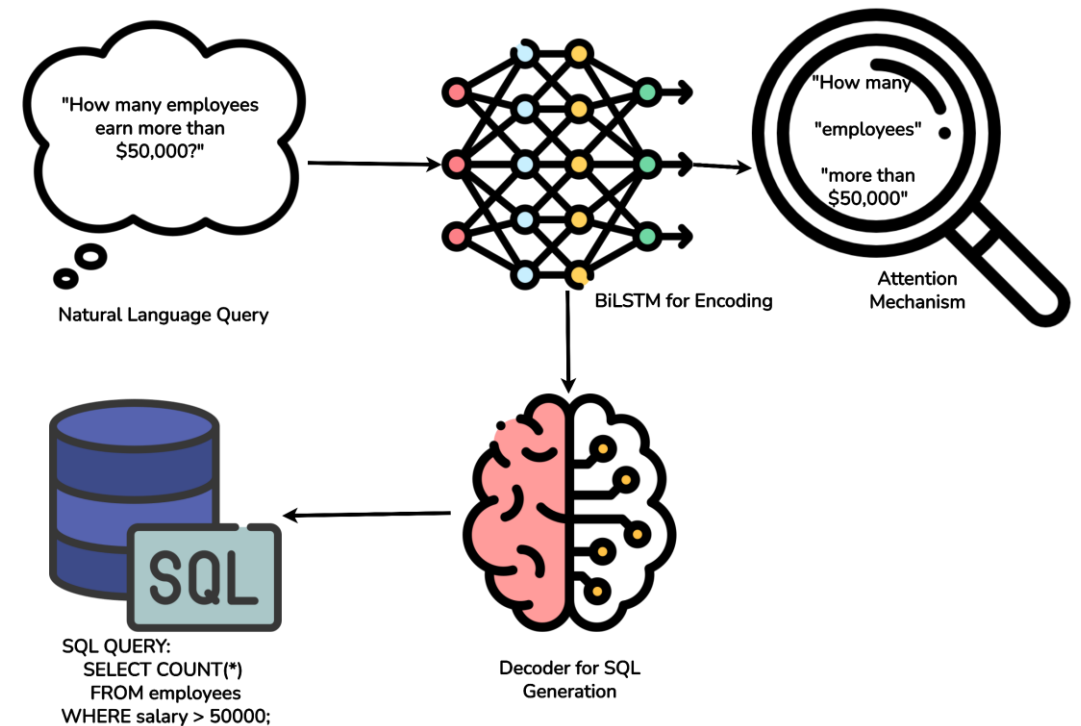


Natural Language Interfaces to Databases (NLIDB)

Background

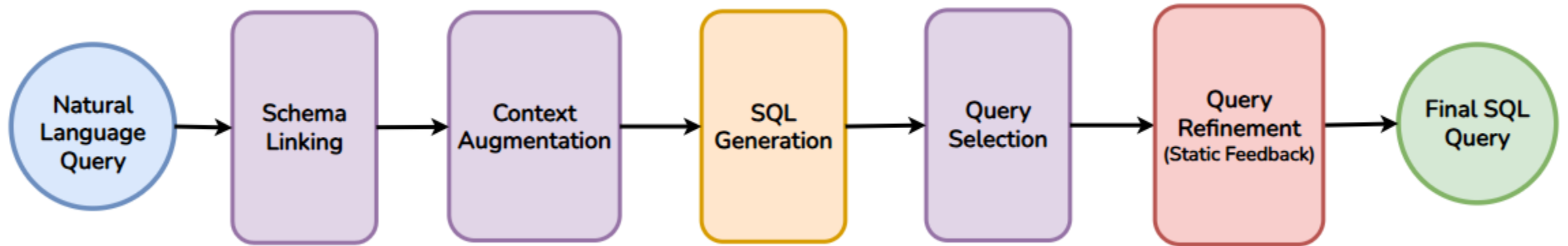


(T. Yu, Z. Li, Z. Zhang, R. Zhang, and D. Radev, "TypeSQL: Knowledge-based Type-Aware Neural Text-to-SQL Generation," 2018) [3]



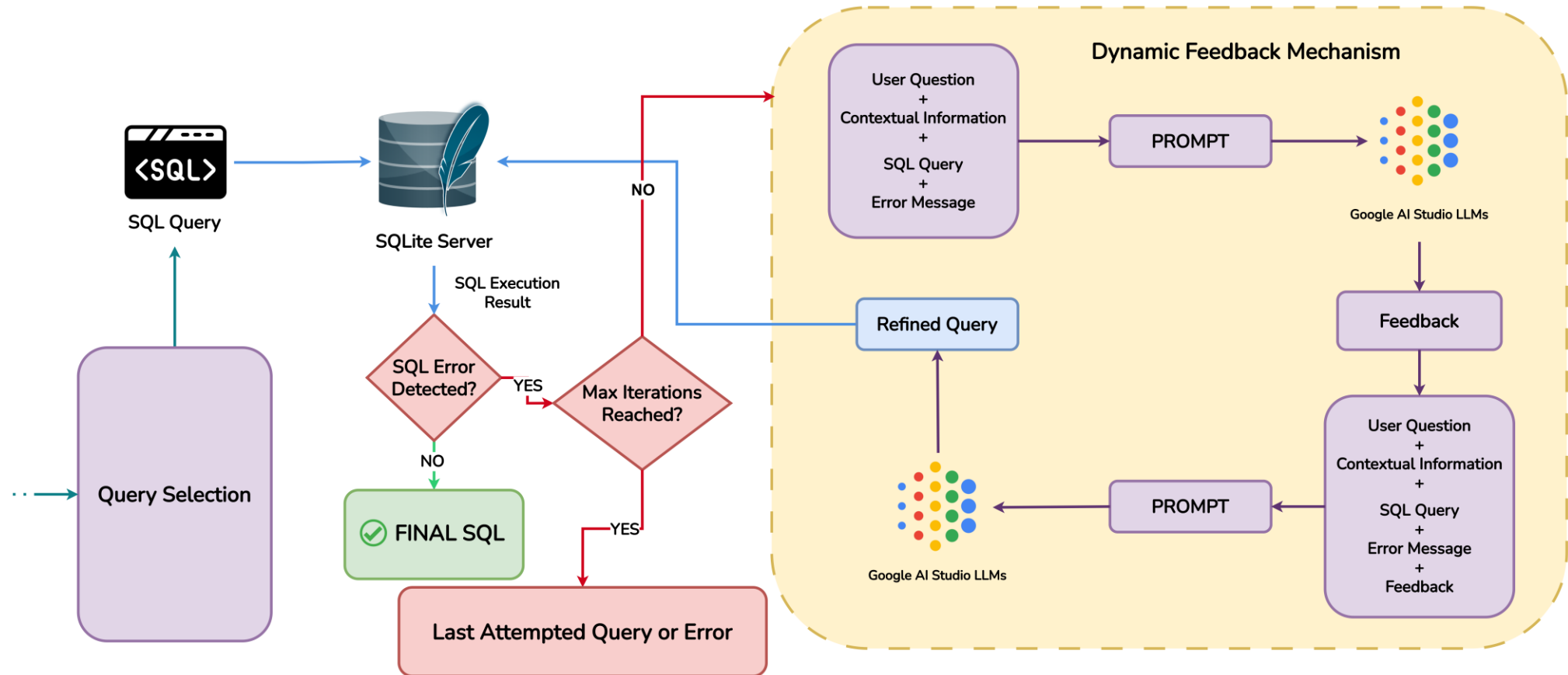
(V. Zhong, C. Xiong, and R. Socher, "Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning," 2017) [2]

Baseline Research



(Z. Cao, Y. Zheng, Z. Fan, X. Zhang, W. Chen, and X. Bai, "RSL-SQL: Robust Schema Linking in Text-to-SQL Generation," 2024) [4]

METHODOLOGY



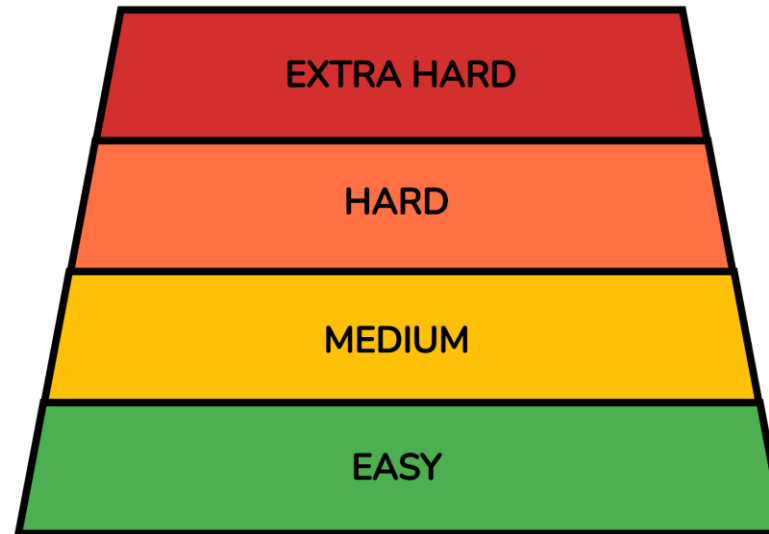
Spider Dataset



70 Databases



2,147 Queries



Multiple joins, Subqueries, and Complex logic

Nested Queries and Aggregations

Joins and basic filters

Simple SELECT Queries (1-2 Tables)

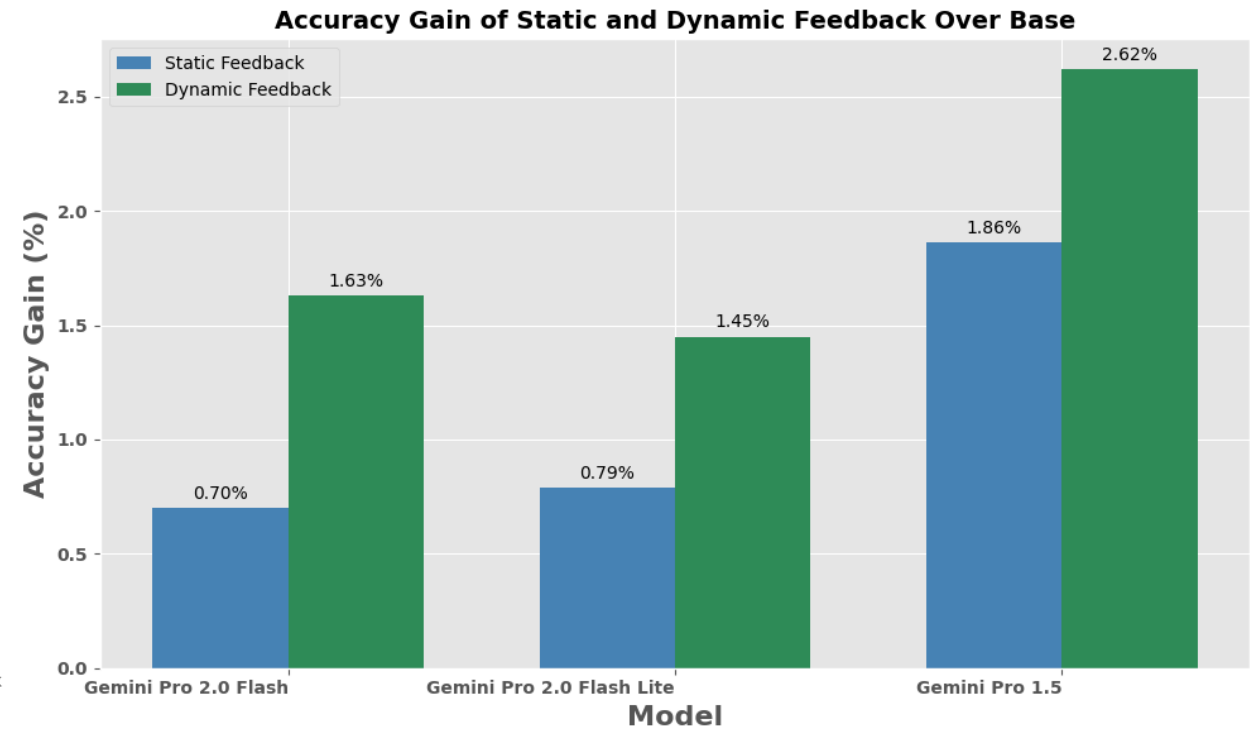
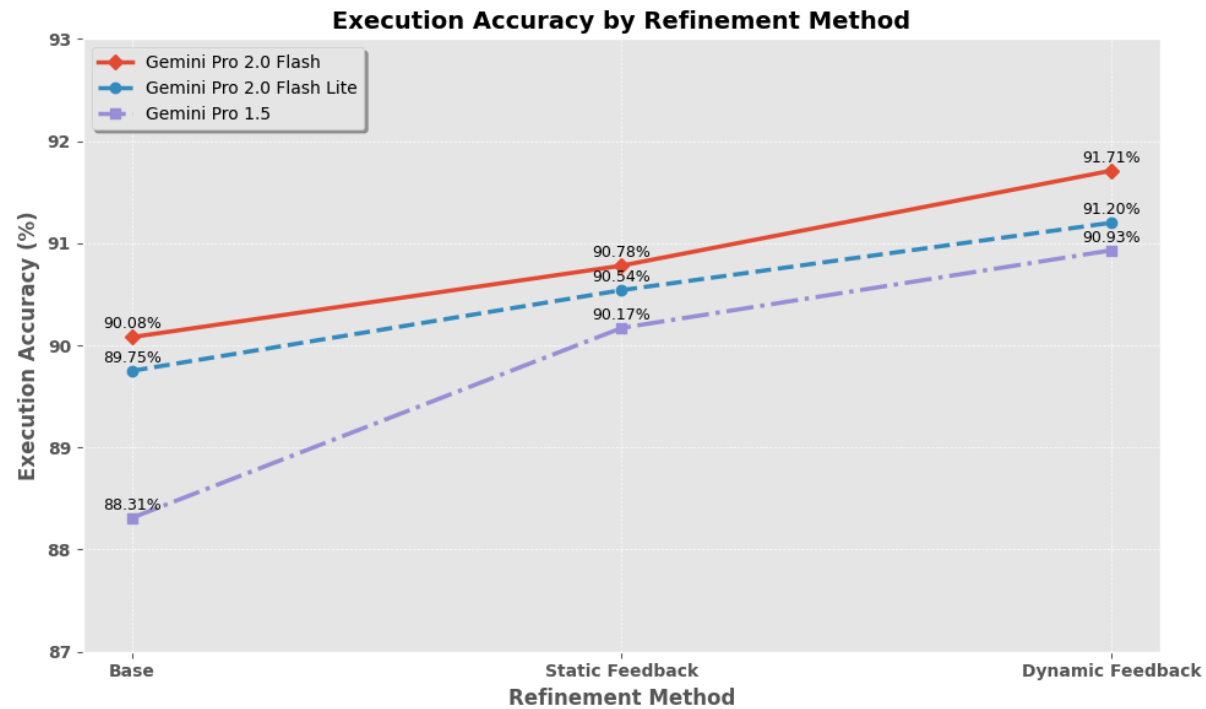
(T. Yu *et al.*, "Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task," 2018) [1]

Example for Dynamic Feedback

```
{
  "id": 74,
  "db": "e_commerce",
  "question": "What are invoices status of all the orders which have not been shipped?",
  "gold_sql": "SELECT invoice_status_code FROM Invoices WHERE invoice_number NOT IN ( SELECT invoice_number FROM Shipments )",
  "simplified_ddl": "#\n# Products(product_id, parent_product_id, product_name, product_price, product_color, product_size, product_weight)\n# Orders(customer_id, product_id, order_date, order_status)\n# Shipments(customer_id, product_id, shipment_date, shipment_status)\n# Customers(customer_id, customer_name, customer_email, customer_phone)\n# Invoices(customer_id, product_id, invoice_date, invoice_status)\n# Payments(customer_id, invoice_id, payment_date, payment_status)\n# Customer_Payment_Methods(customer_id, payment_method_id, payment_method_name)\n# Customer_Review(customer_id, product_id, review_date, review_text, review_rating)\n# Product_Review(product_id, review_date, review_text, review_rating)\n# Product_Review(product_id, review_date, review_text, review_rating)\n# Product_Review(product_id, review_date, review_text, review_rating)",
  "ddl_data": "#\n# Products(product_id[1,2,3],parent_product_id[8,3,1],product_name[Dell monitor,Dell keyboard,iPhone6s],product_price[100,50,1000],product_color[black,white,red],product_size[15.6,13.3,12.5],product_weight[1.5,0.5,0.2])\n# Orders(customer_id[1,2,3],product_id[1,2,3],order_date[2018-01-01,2018-01-01,2018-01-01],order_status[shipped,not shipped])\n# Shipments(customer_id[1,2,3],product_id[1,2,3],shipment_date[2018-01-01,2018-01-01,2018-01-01],shipment_status[shipped,not shipped])\n# Customers(customer_id[1,2,3],customer_name[Dell,Dell,Dell],customer_email[dell@dell.com,dell@dell.com,dell@dell.com],customer_phone[1234567890,1234567890,1234567890])\n# Invoices(customer_id[1,2,3],product_id[1,2,3],invoice_date[2018-01-01,2018-01-01,2018-01-01],invoice_status[shipped,not shipped])\n# Payments(customer_id[1,2,3],invoice_id[1,2,3],payment_date[2018-01-01,2018-01-01,2018-01-01],payment_status[paid,not paid])\n# Customer_Payment_Methods(customer_id[1,2,3],payment_method_id[1,2,3],payment_method_name[credit card,debit card,credit card])\n# Customer_Review(customer_id[1,2,3],product_id[1,2,3],review_date[2018-01-01,2018-01-01,2018-01-01],review_text[good,bad,good],review_rating[5,1,5])\n# Product_Review(product_id[1,2,3],review_date[2018-01-01,2018-01-01,2018-01-01],review_text[good,bad,good],review_rating[5,1,5])\n# Product_Review(product_id[1,2,3],review_date[2018-01-01,2018-01-01,2018-01-01],review_text[good,bad,good],review_rating[5,1,5])\n# Product_Review(product_id[1,2,3],review_date[2018-01-01,2018-01-01,2018-01-01],review_text[good,bad,good],review_rating[5,1,5])",
  "foreign_key": "#\n# Customer_Payment_Methods(customer_id) references Customers(customer_id)\n# Orders(customer_id) references Customers(customer_id)\n# Shipments(customer_id) references Customers(customer_id)\n# Invoices(customer_id) references Customers(customer_id)\n# Payments(customer_id) references Customers(customer_id)\n# Customer_Review(customer_id) references Customers(customer_id)\n# Product_Review(product_id) references Products(product_id)\n# Product_Review(product_id) references Products(product_id)\n# Product_Review(product_id) references Products(product_id)",
  "example": "### Some example pairs of question and corresponding SQL query are provided based on similar problems:\n\n### How to find invoice status of all the orders which have not been shipped?",
  "tables": [
    "Invoices",
    "Orders",
    "Shipments"
  ],
  "columns": [
    "Invoices.invoice_status_code",
    "Orders.order_id",
    "Shipments.order_id",
    "Shipments.invoice_number"
  ],
  "sql_keywords": [
    "INTERSECT"
  ],
  "conditions": [
    "orders have not been shipped"
  ]
},
```

```
{"feedback": "To find invoice statuses for unshipped orders, you need to link Orders, Shipments, and Invoices tables. Use LEFT JOIN to identify orders without corresponding shipments. Then, select the invoice status from the Invoices table for those orders."}
```

Results



REFERENCES

- [1] T. Yu *et al.*, “Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task,” 2018, *arXiv*. doi: 10.48550/ARXIV.1809.08887.
- [2] V. Zhong, C. Xiong, and R. Socher, “Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning,” 2017, *arXiv*. doi: 10.48550/ARXIV.1709.00103.
- [3] T. Yu, Z. Li, Z. Zhang, R. Zhang, and D. Radev, “TypeSQL: Knowledge-based Type-Aware Neural Text-to-SQL Generation,” 2018, doi: 10.48550/ARXIV.1804.09769.
- [4] Z. Cao, Y. Zheng, Z. Fan, X. Zhang, W. Chen, and X. Bai, “RSL-SQL: Robust Schema Linking in Text-to-SQL Generation,” 2024, *arXiv*. doi: 10.48550/ARXIV.2411.00073.
- [5] “xlangai/spider · Datasets at Hugging Face.” Accessed: Feb. 10, 2025. [Online]. Available: <https://huggingface.co/datasets/xlangai/spider>

THANK YOU