

Prompt Engineering and Text to SQL Generation use case

Leah Lu, Sr. Data Scientist

2024 - 06 - 14

Outline

- Prompt Engineering
 - Search vs Prompt
 - In-context learning: Zero-shot, Few-shot prompting
 - Reasoning: Chain of Thought, Reason and Action (ReAct)
 - Connection between Prompt, RAG and AI Agents
- GenAI: Text to SQL Generation
 - Prompt engineering + LLM solution
 - LangChain: prompt engineering + LLM + evaluation
 - Benchmarking: prompt engineering with schema link
 - DSPy: programming prompting
- Prompt engineering live demo
- References

Search vs Prompt nowadays

The screenshot shows a Google search results page for the query "What is Target?". At the top, there's a search bar with the query, followed by a navigation bar with "All", "Images", "Videos", "Shopping", "News", and "More" buttons, and a "Tools" dropdown. Below the search bar is an "AI Overview" section with a red arrow icon and the text: "Target is a mass-market retail corporation that sells a variety of food and general merchandise at discounted prices. Target operates large-scale discount stores and hypermarkets across the United States and around the world, and also offers shopping online and through digital channels. Target's stores are known for their clean, spacious layout and planning, which some say can create an addictive shopping experience." There are three cards below this: one for target.com, one for Wikipedia, and one for Target Corporation. A "Show more" button is visible. At the bottom, there's a "People also ask" section with the question "What is Target mainly known for?" and a dropdown arrow.

- Search at Google

How does Target utilize big data to personalize product recommendations and improve the shopping experience? answer in bullet points and short



- **Customer Purchase History:** Analyzes past purchases to predict future preferences.
- **Browsing Behavior:** Tracks online activity to tailor product suggestions.
- **Demographic Information:** Uses age, location, and other data to customize offers.
- **Behavioral Segmentation:** Groups similar customers to provide relevant recommendations.
- **Predictive Analytics:** Applies machine learning to forecast trends and suggest products.
- **Personalized Marketing:** Sends targeted promotions based on individual profiles.
- **Inventory Management:** Ensures recommended products are available for purchase.

By leveraging these techniques, Target enhances personalization and improves the overall shopping experience.



- Prompt and Response generation at LLM

Prompt Engineering

- Designing and refining the input (prompt) given to an AI model to achieve a desired output, with features Clarity, Specificity, Context, Constraints
- To maximizes the effectiveness and utility of AI-generated responses.

Basic Prompt

"Describe a smartphone."

Engineered Prompt

"Describe the features of the latest high-end smartphone, focusing on its camera capabilities, battery life, screen resolution, and unique selling points."

Prompt Template

- Pre-defined structure or format used to create consistent and effective prompts for AI models, with the benefits of Consistency and Scalability.

```
python
from langchain.prompts import PromptTemplate

recommendation_template = PromptTemplate(
    input_variables=["user_name", "browsing_history", "preferences"],
    template=
        "Based on the browsing history and preferences of {user_name}, provide a list of personalized product recommendations. Browsing history: {browsing_history}. Preferences: {preferences}. "
        "Include reasons why each product is recommended."
)
)

user_name = "John Doe"
browsing_history = "smartphones, laptops, wireless earbuds"
preferences = "tech gadgets, high battery life, budget-friendly"

prompt = recommendation_template.format(
    user_name=user_name,
    browsing_history=browsing_history,
    preferences=preferences
)
print(prompt)
```

Prompt:

"Based on the browsing history and preferences of John Doe, provide a list of personalized product recommendations. Browsing history: smartphones, laptops, wireless earbuds. Preferences: tech gadgets, high battery life, budget-friendly. Include reasons why each product is recommended."

Six strategies from [OpenAI guide](#)

- **Be Specific:** Use clear, detailed prompts to guide the model towards the desired response.
- **Provide Context:** Include relevant background information to help the model understand the request.
- **Use Examples:** Give examples of the desired output to set expectations.
- **Chain of Thought:** Break down complex instructions into smaller, manageable steps.
- **Ask for Clarification:** When responses are unclear, prompt the model to clarify or expand.
- **Iterative Refinement:** Continuously refine prompts based on the model's output to improve results.

Tactic: system message for model persona

Friendly and Casual Assistant

System Message: "You are a friendly and casual assistant. Use a conversational tone and simple language."

User Prompt: "How do I set up a Wi-Fi network at home?"

AI Response: "Hey! Just plug in your router and modem, connect them with a cable, and turn them on. Go to your device's Wi-Fi settings, find your network, and enter the password. You're all set!"

Professional and Technical Assistant

System Message: "You are a professional and technical assistant. Use a formal tone and detailed technical explanations."

User Prompt: "How do I set up a Wi-Fi network at home?"

AI Response: "To set up a Wi-Fi network, connect your router to the modem with an Ethernet cable. Power on both devices and wait for the lights to stabilize. Access your device's network settings, locate the router's SSID, and enter the security key provided."

P Personal / Default project

Dashboard Docs API reference ⚙️ L

Playground Chat Assistants Completions Assistants Fine-tuning Batches Storage Usage API keys

Chat

Unnamed example

gpt-3.5-turbo-16k

SYSTEM

When I ask for help to write something, you will reply with a document that contains at least one joke or playful comment in every paragraph.

USER

Write a thank you note to my steel bolt vendor for getting the delivery in on time and in short notice. This made it possible for us to deliver an important order.

Temperature 1
Maximum Tokens 250
Stop sequences Enter sequence and press Tab
Top P 1
Frequency penalty 0
Presence penalty 0

API and Playground requests will not be used to train our models. [Learn more](#)

Enter user message...

User Add

Parameters impacting generation

Parameter	Function
Temperature	Definition: Controls the randomness and creativity of the output. Interpretation: Higher values produce more random and creative outputs. Lower values produce more deterministic and consistent outputs.
Max Tokens	Definition: Controls the maximum length of the generated text (in tokens). Interpretation: Higher values generate longer outputs. Lower values generate shorter outputs.
Top-p	Definition: Controls the cumulative probability of the candidate word set. Interpretation: Higher values allow more diverse outputs. Lower values produce more predictable outputs.
Frequency Penalty	Definition: Controls the penalty for frequent words to reduce repetition. Interpretation: Higher values reduce repetition. Lower values allow more repetition.
Presence Penalty	Definition: Controls the penalty for new words to reduce repetition. Interpretation: Higher values encourage new words. Lower values allow more repetition.

``` What is the risk for AI? ```

- # High Parameter Settings (Diverse and Detailed Output)
- *AI poses risks such as significant job displacement, raising ethical and privacy concerns, and exacerbating biases within algorithms. Additionally, AI technologies can be misused for malicious purposes, including cyberattacks and surveillance. The rapid advancement of AI without proper regulation could lead to unforeseen consequences affecting society at large.*
- # Low Parameter Settings (Concise and Predictable Output)
- *AI risks include job loss and ethical issues. There are also concerns about bias in AI algorithms. Misuse of AI technology is another significant risk.*
- # Moderate Parameter Settings (Balanced Output)
- *AI risks include job displacement and ethical concerns related to privacy and decision-making. There is also a risk of inherent biases in AI systems affecting fairness. Additionally, the misuse of AI for harmful purposes poses significant dangers.*

In-context learning: Zero and few -shot examples

Prompt: ""Translate the following English sentence to Chinese: 'The weather is nice today. ' ""

Response: "今天的天气很好。"

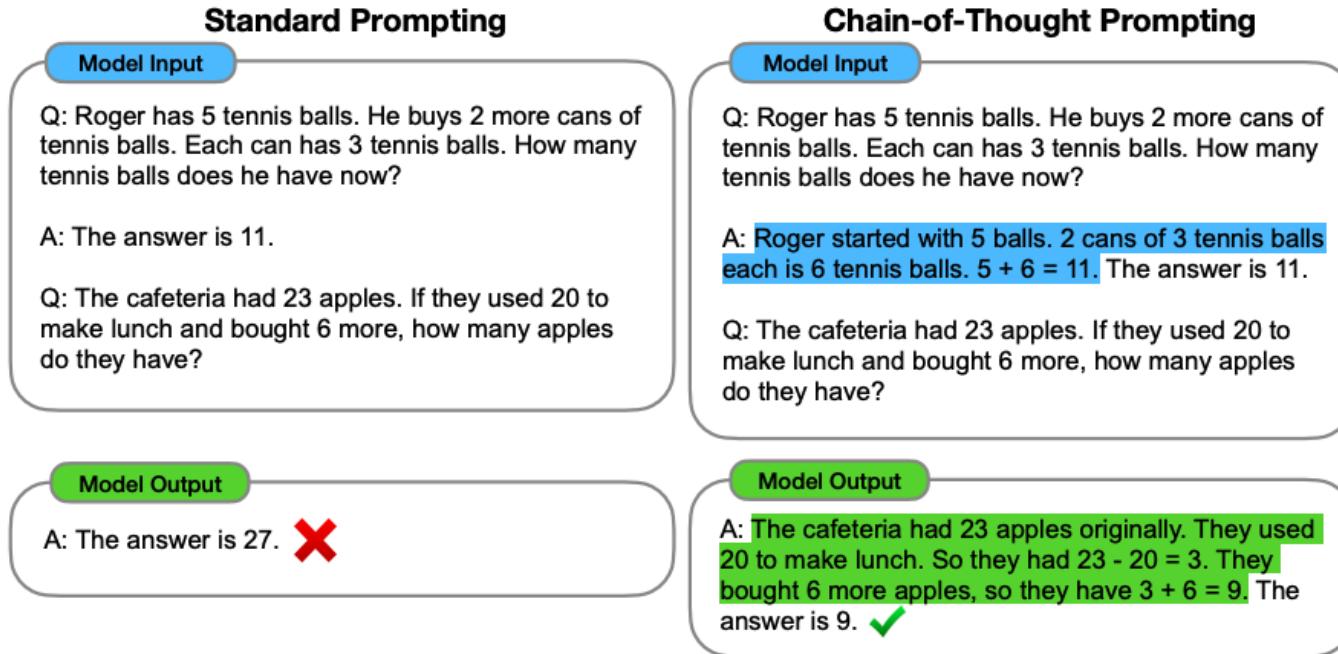
Prompt: "**Translate the following English sentences to Chinese:**

1. 'Good morning.' -> '早上好。'
2. 'How are you?' -> '你好吗？'
3. 'I would like a coffee.' -> '我想要一杯咖啡。'

Translate the following English sentence to Chinese: 'The weather is nice today. ''

Response: "今天的天气很好。"

Chain of Thought (CoT) prompting



LLMs think step by step and "Reasoning"

(a) Few-shot

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

(Output) The answer is 8. X

(b) Few-shot-CoT

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are $16 / 2 = 8$ golf balls. Half of the golf balls are blue. So there are $8 / 2 = 4$ blue golf balls. The answer is 4. ✓

(c) Zero-shot

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: The answer (arabic numerals) is

(Output) 8 X

(d) Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: **Let's think step by step.**

(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓

Reason and Action (ReAct): "Reasoning" and "Acting"

"""Aside from the Apple Remote, what other devices can control the program?"""

(1d) ReAct (Reason + Act)

Thought 1: I need to search Apple Remote and find the program it was originally designed to interact with.

Act 1: Search[Apple Remote]

Obs 1: The Apple Remote is a remote control introduced in October 2005 by Apple ... originally designed to control the Front Row media center program ...

Thought 2: Apple Remote was originally designed to control the Front Row media center program. I need to search Front Row next and find what other device can control it.

Act 2: Search[Front Row]

Obs 2: Could not find [Front Row]. Similar: ['Front Row Seat to Earth', 'Front Row Motorsports', 'Front Row (software)', ...]

Thought 3: Front Row is not found. I need to search Front Row (software).

Act 3: Search[Front Row (software)]

Obs 3: Front Row is a discontinued media center software ...

Thought 4: Front Row (software) is controlled by an Apple Remote or the keyboard function keys. So the answer is keyboard function keys.

Act 4: Finish[keyboard function keys]



- ReAct elicits LLMs to reason such as CoT and act, which is to interact with external tools to lead to more reliable and factual responses.
- LangChain leverages the ReAct framework to build AI agents

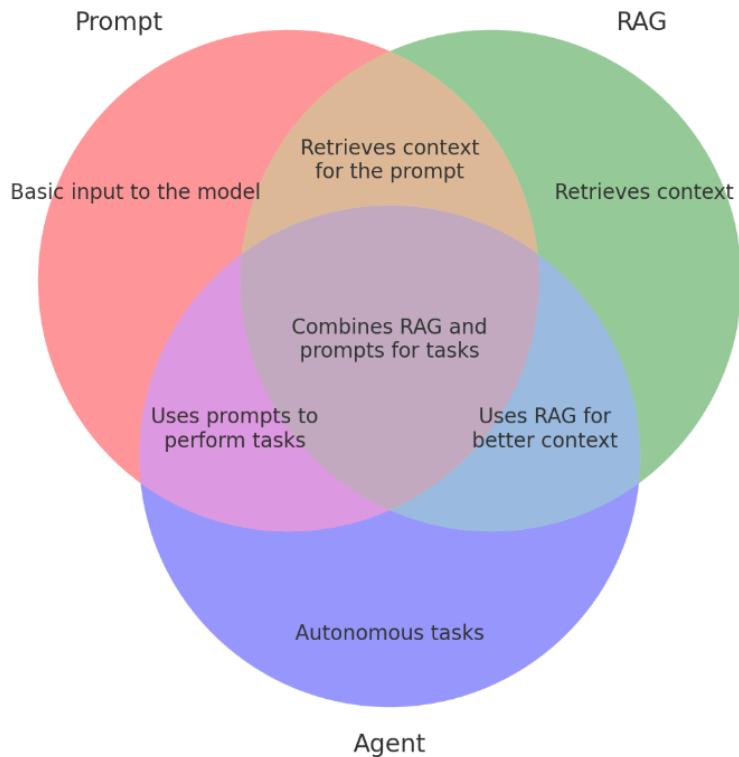
<https://www.promptingguide.ai/techniques/react>

AI Agent – Jarvis (Just A Rather Very Intelligent System)



- Designed to achieve specific goals, capable of performing complex tasks, interacting with users, and adapting to new situations
- Agents often use prompts and other techniques to enhance their capabilities

Connection between Prompt, RAG and Agent



- **Prompt:** The input (prompt) given to an AI model to achieve a desired output
- **RAG:** Builds on prompts by adding a retrieval component that enhances the context and information available for generating responses.
- **AI Agent:** Utilizes prompts (and often RAG) to autonomously perform tasks, interact with users, and achieve specific goals.

GenAI: Text to SQL Generation



- "List the names of employees who work in the marketing department."
 - "Show the total sales amount for each product category."
 - "Find all orders placed by customer John Doe."
-
- SELECT name FROM employees WHERE department = 'Marketing';
 - SELECT category, SUM(sales_amount) FROM sales GROUP BY category;
 - SELECT * FROM orders WHERE customer_name = 'John Doe';

*Note: Answering the questions accurately is not required, but the SQL query generation

Terminology

- **Prompt:** an input or instruction given to an AI model to elicit a desired response
- **In-context learning (ICL):** LLMs learn and adapt to new tasks based on the context provided within a single prompt, which can include zero-shot, few-shot, or many-shot examples
- **LangChain:** a framework for developing applications using language models
- **HuggingFace:** open-source libraries and platforms for large models and datasets
- **Text to SQL generation:** refers to the process of converting natural language queries into SQL queries using language models
- **Schema Link:** refers to the process of associating parts of a natural language query with the corresponding elements in a database schema
- **DSPy:** a framework for algorithmically optimizing LM prompts and weights

How to evaluate?

Metric	Description	Example
Exact Matching (EM)	Checks if the predicted SQL is exactly the same as the gold SQL.	Gold SQL: SELECT name FROM students WHERE age > 18 Predicted SQL: SELECT name FROM students WHERE age > 18 Result: Match (EM = 1)
Execution Accuracy (EX)	Measures if the result of the predicted SQL matches the gold result.	Gold SQL Result: [Alice, Bob] Predicted SQL Result: [Alice, Bob] <i>Result: Match (EX = 1)</i>

Public Datasets for Text to SQL

Dataset	Tables	Question-SQL Pairs	Domains	Key Features
WikiSQL	25,000	80,000	Various	Large-scale, cross-domain, low SQL complexity.
Spider	--	10,181	138	Multi-table, complex SQLs, cross-domain, variants include Spider-Realistic (no explicit column names), Spider-Syn (tests adaptability to paraphrases), Spider-DK (tests robustness with domain knowledge).
BIRD	95	12,751	--	Large-scale, cross-domain, complex SQLs, SQL efficiency.

Reference: Zhang B, Ye Y, Du G, Hu X, Li Z, Yang S, Liu CH, Zhao R, Li Z, Mao H. Benchmarking the Text-to-SQL Capability of Large Language Models: A Comprehensive Evaluation. arXiv preprint arXiv:2403.02951. 2024 Mar 5.

Spider data set



Spider 1.0
Yale Semantic Parsing and Text-to-SQL Challenge

What is Spider?

Feb. 5th, 2024: We will no longer accept submissions for Spider 1.0 evaluations or update its leaderboard. Look forward to the release of Spider 2.0, a more realistic and challenging benchmark in the era of LLMs, expected this

March June. Stay tuned!

Spider is a large-scale *complex and cross-domain* semantic parsing and text-to-SQL dataset annotated by 11 Yale students. The goal of the Spider challenge is to develop natural language interfaces to cross-domain databases. It consists of 10,181 questions and 5,693 unique complex SQL queries on 200 databases with multiple tables covering 138 different domains. In Spider 1.0, different complex SQL queries and databases appear in train and test sets. To do well on it, systems must *generalize well to not only new SQL queries but also new database schemas*.

Why we call it "Spider"? It is because our dataset is complex and cross-domain like a spider crawling across multiple complex (with many foreign keys) nests (databases).

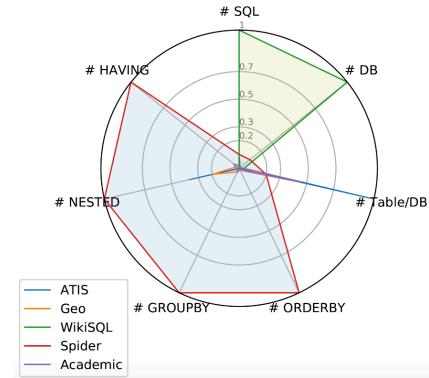
XLang Lab for building LM agents!

Leaderboard - Execution with Values

Our current models do not predict any value in SQL conditions so that we do not provide execution accuracies. However, we encourage you to provide it in the future submissions. For value prediction, your model should be able to 1) copy from the question inputs, 2) retrieve from the database content (database content is available), or 3) generate numbers (e.g. 3 in "LIMIT 3"). *Notice:* Test results after May 02, 2020 are reported on the new release (collected some annotation errors).

Rank	Model	Test
1 Nov 2, 2023	MiniSeek <i>Anonymous</i> Code and paper coming soon	91.2
1 Aug 20, 2023	DAIL-SQL + GPT-4 + Self-Consistency <i>Alibaba Group</i> (Gao and Wang et al.,'2023) code	86.6
2 Aug 9, 2023	DAIL-SQL + GPT-4 <i>Alibaba Group</i> (Gao and Wang et al.,'2023) code	86.2
3 October 17, 2023	DPG-SQL + GPT-4 + Self-Correction <i>Anonomous</i>	85.6

Why Spider?



Some examples look like the following:

Easy

What is the number of cars with more than 4 cylinders?

```
SELECT COUNT(*)
FROM cars_data
WHERE cylinders > 4
```

Medium

For each stadium, how many concerts are there?

```
SELECT T2.name, COUNT(*)
FROM concert AS T1 JOIN stadium AS T2
ON T1.stadium_id = T2.stadium_id
GROUP BY T1.stadium_id
```

BIRD-SQL dataset



BIRD-SQL

A Big Bench for Large-Scale Database Grounded Text-to-SQLs



How many accounts are eligible for loans in New York City?

External Knowledge:
The condition of loans is that the type of the account should be "OWNER".

```
SELECT COUNT(*) FROM account  
WHERE account.type = "OWNER" AND disp_id = "NY";
```



About BIRD

Page Views 50537

BIRD (Big Bench for LaRge-scale Database Grounded Text-SQL Evaluation) represents a pioneering, cross-domain dataset that examines the impact of extensive database contents on text-to-SQL parsing. BIRD contains over **12,751** unique question-SQL pairs, **95** big databases with a total size of **33.4 GB**. It also covers more than **37** professional domains, such as blockchain, hockey, healthcare and education, etc.

Paper

Code

Mini-Dev (500)

Train Set

🔥 Dev Set

Leaderboard - Execution Accuracy (EX)

	Model	Code	Size	Oracle Knowledge	Dev (%)	Test (%)
	Human Performance <i>Data Engineers + DB Students</i>			✓		92.96
1 May 14, 2024	ExSL + granite-20b-code <i>IBM Research AI</i>		20B	✓	65.38	67.86
2 May 31, 2024	Byte-SQL <i>ByteDance ByteBrain</i>		33B	✓	64.54	67.75
3 May 21, 2024	CHESS <i>Stanford [Talaei et al.'24]</i>		UNK	✓	65.00	66.69
4 Jan 14, 2024	MCS-SQL + GPT-4 <i>Dunamu</i>		UNK	✓	63.36	65.45
5 Apr 08, 2024	OpenSearch-SQL,v1 + GPT-4 <i>Alibaba Cloud</i>		UNK	✓	61.34	64.95

Open dataset Spider at Huggingface

Datasets: xlangai/spider

Tasks: Text2Text Generation Languages: English Multilinguality: monolingual Size Categories: 1K< n <10K Languages: English

Annotations Creators: expert-generated Source Datasets: original ArXiv: arxiv:1809.08887 Tags: text-to-sql Croissant

Dataset card Viewer Files and versions Community 6

Dataset Viewer

Split (2)
train · 7k rows

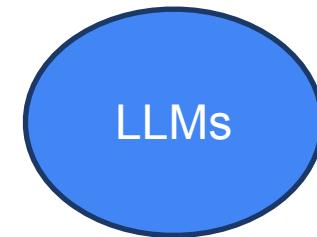
Search this dataset

db_id	query	question	query_toks
string · classes	string · lengths	string · lengths	sequence · lengths
140 values	18 577	16 224	4 90
department_management	SELECT count(*) FROM head WHERE age > 56	How many heads of the departments are older than...	["SELECT", "count", "(", "*", ")"]
department_management	SELECT name , born_state , age FROM head ORDER BY age	List the name, born state and age of the heads of...	["SELECT", "name", ".", "born_state", ".", "age", ...]
department_management	SELECT creation , name , budget_in_billions FROM...	List the creation year, name and budget of each...	["SELECT", "creation", "", "name", "", ...]
department_management	SELECT max(budget_in_billions) ,...	What are the maximum and minimum budget of the...	["SELECT", "max", "(", "budget_in_billions", ")"]
department_management	SELECT avg(num_employees) FROM department WHERE rankin...	What is the average number of employees of the...	["SELECT", "avg", "(", "num_employees", ")"]

- Dataset snippets
- {question + db-id : query}
- Huggingface link: <https://huggingface.co/datasets/xlangai/spider>

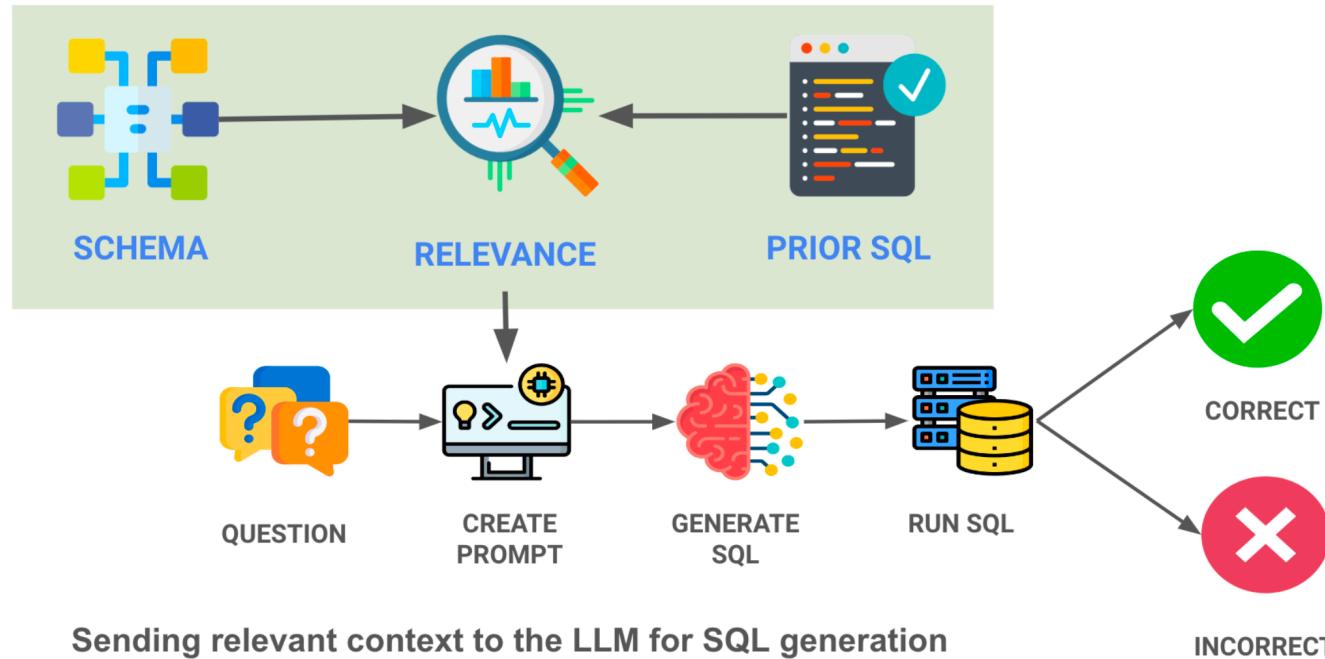
What the challenges are:

- Ambiguity in Natural Language
- Complex Queries
- Schema Understanding
- Synonyms and Variations
- Contextual Understanding
- Handling Errors and Edge Cases
- Performance and Scalability
- Natural Language Diversity



- Deep understanding of natural language
- Context learning
- Handle complexity and diversity

Prompt Engineering + LLMs solution

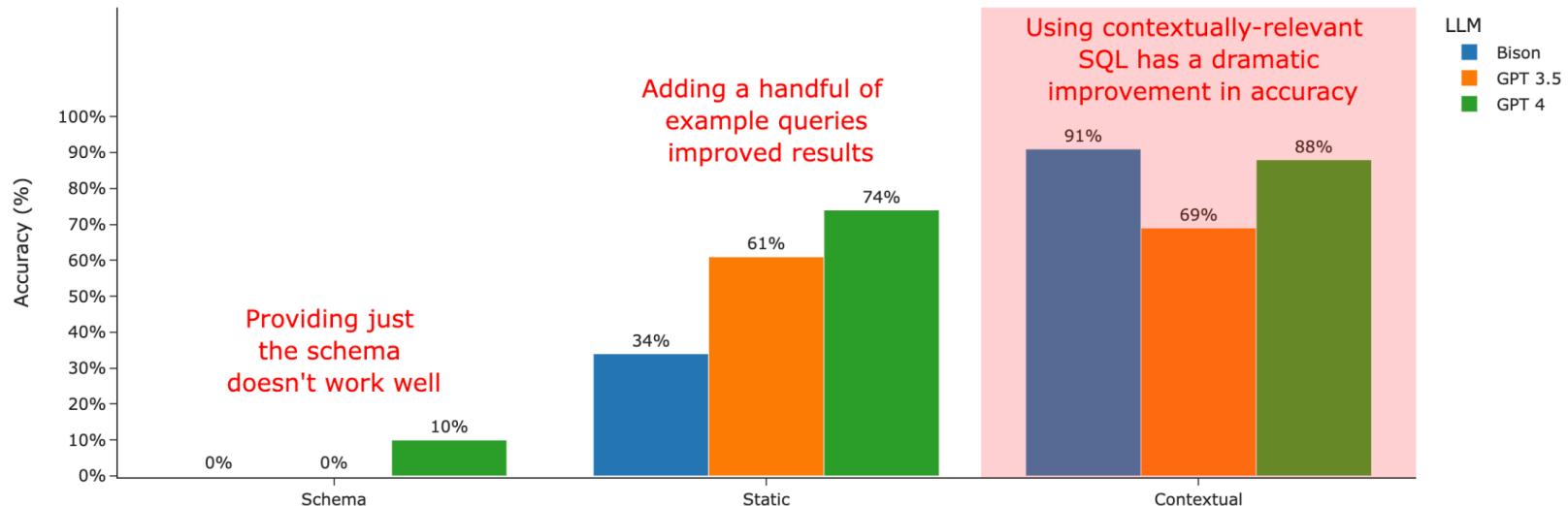


Factors impacting the performance

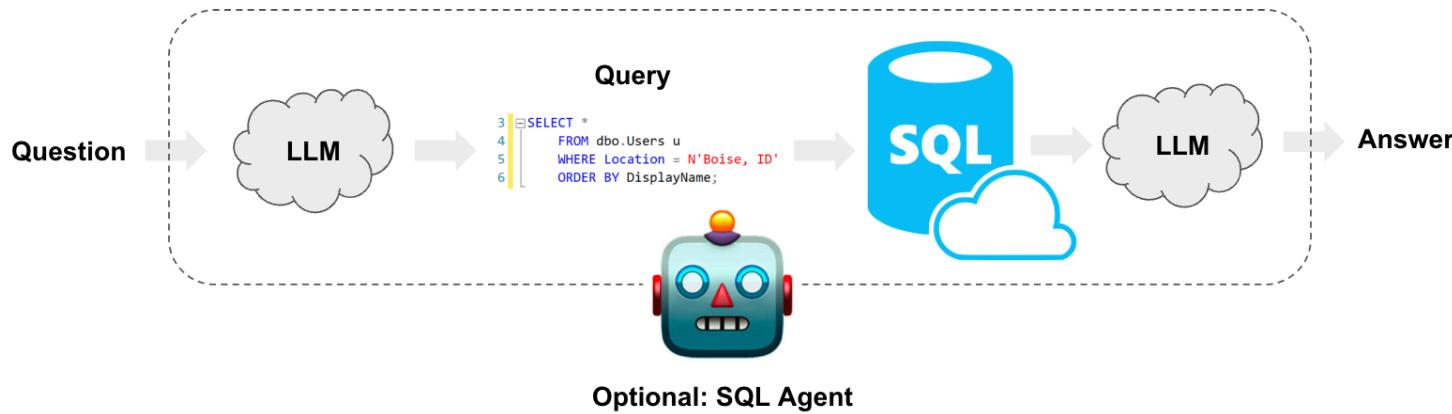
Factors	Description
Model Architecture	Transformer-based models (e.g., GPT-4, GPT-3.5, Llama) with pre-training and fine-tuning.
Natural Language Understanding	Recognizing user intent and resolving ambiguities.
Schema Linking	Mapping natural language entities to database columns/tables; understanding database context.
SQL Knowledge	Generating syntactically correct and semantically meaningful SQL queries; optimizing performance.
User Interaction	Feedback mechanisms and interactive query refinement.

Observation (with skeptics)

SQL Accuracy using Contextual Examples



LangChain: Text to SQL tutorial



High level steps:

1. Convert question to **SQL query**: Model converts user input to a SQL query.
2. Execute **SQL query**: Execute the SQL query.
3. Answer the question: Model responds to user input using the query results.

```
from langchain_community.utilities import SQLDatabase
```

```
db = SQLDatabase.from_uri("sqlite:///Chinook.db")
print(db.dialect)
```

```
from langchain_core.output_parsers import StrOutputParser
from langchain_core.prompts import ChatPromptTemplate

system = """Double check the user's {dialect} query for common mistakes, including:
- Using NOT IN with NULL values
- Using UNION when UNION ALL should have been used
- Using BETWEEN for exclusive ranges
- Data type mismatch in predicates
- Properly quoting identifiers
- Using the correct number of arguments for functions
- Casting to the correct data type
- Using the proper columns for joins"""

If there are any of the above mistakes, rewrite the query. If there are no mistakes, just reproduce the original query.
```

```
Output the final SQL query only."""
prompt = ChatPromptTemplate.from_messages(
    [("system", system), ("human", "{query}")]
).partial(dialect=db.dialect)
validation_chain = prompt | llm | StrOutputParser()

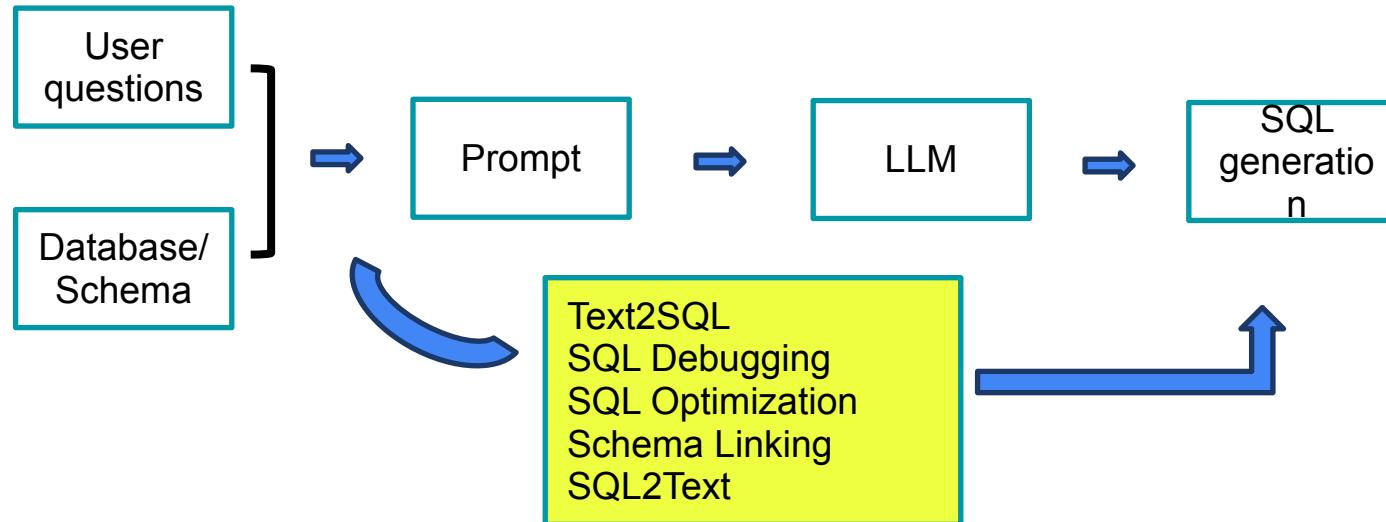
full_chain = {"query": chain} | validation_chain
```

```
# SQL query validation:
db.run(SQLquery)
```

- Database: provided
- Schema: provided
- Question: "What's the average Invoice from an American customer whose Fax is missing since 2003 but before 2010"
- SQLquery: "SELECT
AVG(Invoice.Total) AS
AverageInvoice\nFROM Invoice\nJOIN
Customer ON Invoice.CustomerId =
Customer.CustomerId\nWHERE
Customer.Country = 'USA'\nAND
Customer.Fax IS NULL\nAND
Invoice.InvoiceDate >=
'2003-01-01'\nAND Invoice.InvoiceDate
< '2010-01-01'"

Benchmarking the Text-to-SQL Capability of Large Language Models: A Comprehensive Evaluation

Bin Zhang^{1,2,3,*} Yuxiao Ye^{1,4,*} Guoqing Du^{1,*} Xiaoru Hu^{1,*} Zhishuai Li¹ Sun Yang¹
Chi Harold Liu⁴ Rui Zhao¹ Ziyue Li¹ Hangyu Mao^{1,✉}



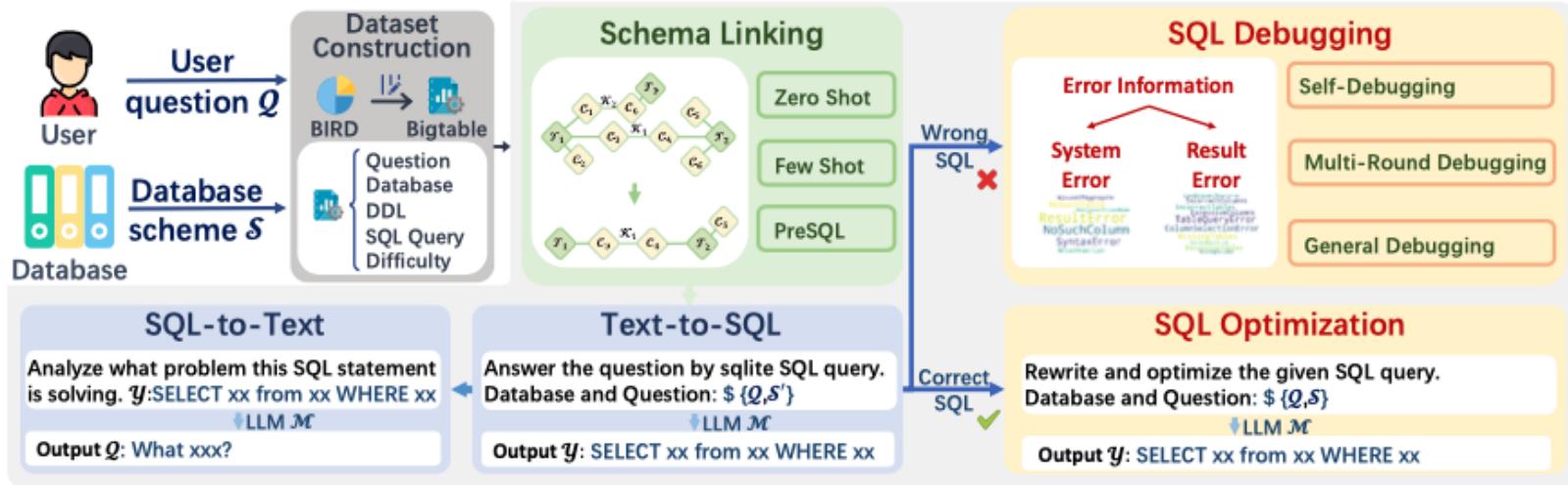


Figure 1: Benchmarking tasks in Text-to-SQL pipeline.

Summaries from the paper

- Pros:
 - Conducted a systematic benchmarking of the various sub-tasks within the Text-to-SQL pipeline, encompassing SQL-Debugging, SQL-Optimization, SQL-to-Text, and Schema-Linking
 - Utilized optimal prompt templates for each sub-task and provided prompt templates
 - Evaluated the performance across 6 LLMs (ChatGPT, SQLCoder, InternLM, CodeLLama, InternLM2, Llama2)
- Cons:
 - The paper appeared in arXiv since March 2024, still under peer-review?
 - The performance is primarily evaluated in the 200 questions and answers pairs, which makes it skeptical for robustness in the evaluation results

Schema Linking

Question: Find the **names** of **schools** that have a **donation** with **amount** above **8.5**.

Schema:

school		endowment	
id	name	location	id
			school_id

foreign key

SQL `select t2.name`

Query: `from endowment as t1 join school as t2
on t1.school_id = t2.id
where t1.amount > 8.5`

Show the **ids** and **details** of the **investors** who have a **transaction** with **type code** "sale".

investor

transaction

investor		transaction		
id	details	id	investor_id	type_code
				foreign key

`select t1.id , t1.details`

`from investors as t1 join transactions as t2
on t1.id = t2.investor_id
where t2.type_code = "sale"`

- Understand the Database Schema including tables, columns and Relationships
- Parse the Natural Language Query to better understand the request
- Identify Schema Elements in the Query

Prompt template and evaluation results

```

1 ### Answer the question by sqlite SQL query only and with no
   ↴ explanation
2 ### Sqlite SQL tables, with their properties:
3 #
4 # stadium(Stadium_ID,Location,Name,Capacity,Highest,Lowest,Average);
5 # singer(Singer_ID,Name,Country,Song_Name,Song_release_year,Age,
   ↴ Is_male);
6 # concert(concert_ID,concert_Name,Theme,Stadium_ID,Year);
7 # singer_in_concert(concert_ID,Singer_ID).
8 #
9 ### How many singers do we have?
10 ### SQL:

```

Listing 4: Proposed prompt template “SimpleDDL-MD-Chat”.

Table 3: EX (%) of 8 zero-shot prompt templates on Spider dev set.

prompt template	SQLCoder-34B	Codellama-34B	InternLM-70B	Llama2-Chat-70B
DDL-HTML-Chat	57.8	63.7	65.0	49.6
DDL-HTML-Complete	61.8	65.2	53.8	50.2
DDL-MD-Chat	63.2	68.4	66.3	48.7
DDL-MD-Complete	62.4	69.8	64.1	46.8
DDL-Coding-Chat	60.3	67.1	66.1	48.4
DDL-Coding-Complete	59.7	66.9	62.9	53.4
SimpleDDL-MD-Chat	65.0	71.6	67.4	54.7
SimpleDDL-MD-Complete	63.3	66.0	61.7	50.2

```

1 DDL Format:
2 CREATE TABLE stadium (stadium_id NUMBER PRIMARY KEY, location TEXT,
   ↴ name TEXT, capacity NUMBER, highest NUMBER, lowest NUMBER, average
   ↴ NUMBER);
3 SimpleDDL Format:
4 stadium(Stadium_ID,Location,Name,Capacity,Highest,Lowest,Average);

```

Listing 1: Examples of “DDL” and “SimpleDDL” formats.

```

1 MD Format:
2 ### How many singers do we have?
3 HTML Format:
4 <Question>How many singers do we have?</Question>
5 Coding Format:
6 /* How many singers do we have? */

```

Listing 2: Examples of “MD”, “HTML” and “Coding” formats.

```

1 Complete Format:
2 Question: How many singers do we have?
3 SQL: SELECT
4 Chat Format:
5 Question: How many singers do we have?
6 SQL:

```

Listing 3: Examples of “Complete” and “Chat” formats.

```

1  ### Rewrite and optimize the given SQL query to improve SQL query
   ↴ efficiency and minimize SQL execution time while ensuring
   ↴ correctness. Only output sql query, do not output any other content.
   ↴ Only output sql query, do not output any other content.
2  ### Here are some reference cases:
3  #
4  # Question: List out the age of users who located in Vienna, Austria
   ↴ obtained the badge?
5  # SQL Query: SELECT Age FROM users WHERE Location = 'Vienna, Austria'
   ↴ AND Id IN (SELECT UserId FROM badges)
6  # New SQL Query: SELECT u.Age FROM users AS u INNER JOIN badges AS b
   ↴ ON u.Id = b.UserId WHERE u.Location = 'Vienna, Austria'
7  # Explanation: By applying a JOIN operation instead of a subquery with
   ↴ IN can improve efficiency, as the database may execute the JOIN and
   ↴ filtering processes concurrently in just one operation without the
   ↴ need to store the intermediate results to filter primary query.
8  #...
9  #
10 #### Sqlite SQL tables, with their properties:
11 #
12 #{SimpleDDLdd}
13 #
14 #### Question: What is the short name and fifa ID for Queens Park
   ↴ Rangers Football Team?...
15 #### SQL Query:SELECT team_short_name , team_fifa_api_id FROM Team
   ↴ WHERE team_long_name = "Queens Park Rangers"
16 #### New SQL Query:

```

Listing 13: Prompt template “w/ demo + comments” for SQL Optimization.

```
1   ### Write the correct SQLite SQL Query corresponding to the Question  
2   ↳ based on the database, the Wrong SQL Query and the cause of the  
3   ↳ error.  
4   ### Sqlite SQL tables, with their properties:  
5   #  
6   #{SimpleDDL}  
7   #  
8   ### Question: Under whose administration does the school with the  
9   ↳ highest number of test takers whose total SAT Scores are greater or  
10   ↳ equal to 1500 belong to? Indicate his or her full name.  
11   ### Wrong SQL Query:  
12   SELECT T1.AdmFName1 , T1.AdmLName1 FROM schools AS T1 JOIN satscores  
13   ↳ AS T2 ON T1.CDSCode = T2.cds WHERE T2.NumTstTakr = ( SELECT  
14   ↳ NumTstTakr FROM satscores GROUP BY cds HAVING NumGE1500 >= 1500  
15   ↳ ORDER BY NumTstTakr DESC LIMIT 1 )  
16   ### Error Information:  
17   Executed correctly, but with the wrong result.  
18   You have found the correct tables. But you select wrong columns, you  
19   ↳ need to select more Columns.  
20   ### Correct SQL:
```

Listing 12: Prompt template “w/ Wrong SQL + All_error_info + Comment” for SQL Debugging.

```
1 Given the database schema and question, perform the following actions:  
2 1 - Rank all the tables based on the possibility of being used in the  
↳ SQL according to the question from the most relevant to the least  
↳ relevant, Table or its column that matches more with the question  
↳ words is highly relevant and must be placed ahead.  
3 2 - Check whether you consider all the tables.  
4 3 - Output a list object in the order of step 2, Your output should  
↳ contain all the tables. The format should be like:  
5 [  
6     "table_1", "table_2", ...  
7 ]  
8  
9 Database schemas with their properties:  
10 {SimpleDDL}  
11  
12 Question: What is the short name and fifa ID for Queens Park Rangers  
↳ Football Team?In the database, short name of the football team  
↳ refers to team_short_name; Queens Park Rangers refers to  
↳ team_long_name = 'Queens Park Rangers';fifa ID refers to  
↳ team_fifa_api_id.  
13 Answer(Only output the list object containing all tables, do not  
↳ output other content):
```

Listing 16: Prompt template for zero-shot schema linking.

```

1 Given the database schema and question, perform the following actions:
2 1 - Evaluate the importance of each table **in relation to the SQL
   ↴ query**, prioritizing tables and columns that closely match the
   ↴ question words. Rank the tables from the most crucial to the least
   ↴ crucial.
3 2 - Focus on identifying and listing only the most important tables
   ↴ based on the evaluation in step 1.
4 3 - Output a list object representing the order determined in step 2.
   ↴ The output should include **the most important tables** and follow
   ↴ this format:
5 [
6     "most_important_table_1", "most_important_table_2", ...
7 ]
8
9 Schema:
10 # department(Department_ID,Name,Creation, ...)
11 # head(head_ID,name,born_state,age)
12 # management(department_ID,head_ID,temporary_acting)
13 Foreign key:
14 management(department_ID) REFERENCES department(Department_ID) \
   ↴ \management(head_ID) REFERENCES head(head_ID)
15 Question: what are the distinct creation years of the departments
   ↴ managed by a secretary born in state 'Alabama'?
16 Answer: ["department","management","head"]
17
18 Schema:
19 # Country(id,name)
20 # League(id,country_id,name)
21 # Player(id,player_api_id,player_name,player_fifa_api_id,birthday,
   ↴ \height,weight)
22 # Player_Attributes(id,player_fifa_api_id, ...)
23 # Team(id,team_api_id,team_fifa_api_id,team_long_name,team_short_name)
24 # Team_Attributes(id,team_fifa_api_id, ...)
25 # sqlite_sequence(name,seq)
26 Foreign key:
27 Player_Attributes(player_api_id) REFERENCES Player(player_api_id) \
   ↴ \nleague(country_id) REFERENCES country(id)\nTeam_Attributes(
   ↴ \team_api_id) REFERENCES Team(team_api_id)\nMatch(away_player_11)
   ↴ REFERENCES Player(player_api_id)
28 Question: List the names of all left-footed players who have overall
   ↴ rating between 85 and 90.
29 Answer: ["Player","Player_Attributes"]
30
31 Database schemas with their properties:
32 {SimpleDDL}
33
34 Question: What is the short name and fifa ID for Queens Park Rangers
   ↴ Football Team? In the database, short name of the football team
   ↴ refers to team_short_name; Queens Park Rangers refers to
   ↴ team_long_name = 'Queens Park Rangers';fifa ID refers to
   ↴ team_fifa_api_id.
35 Answer:

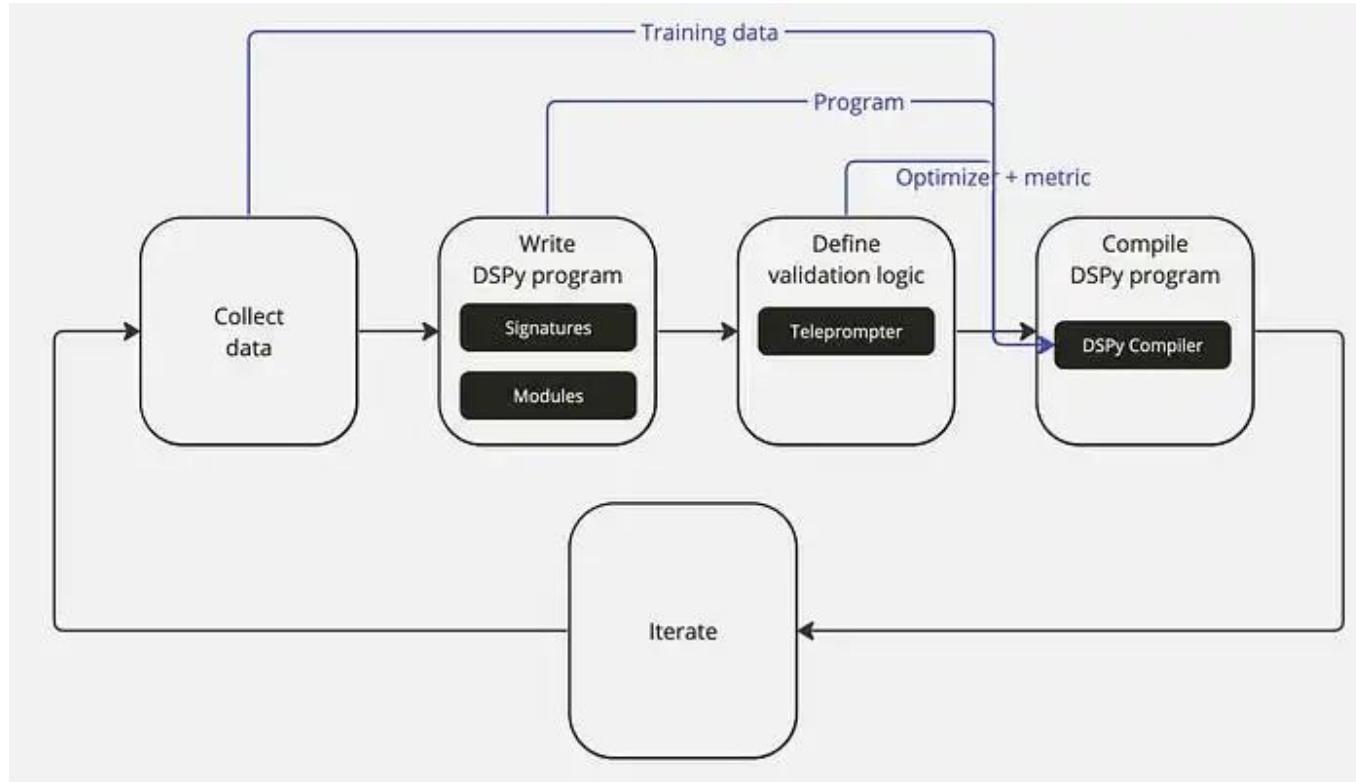
```

Listing 17: Prompt template for few-shot schema linking.

DSPy: Programming—not prompting—Foundation Models



- DSPy is a open-source framework for algorithmically optimizing LM prompts and weights from Stanford NLP group
- DSPy github: <https://github.com/stanfordnlp/dspy>



Replace how to prompt the LM with what a transformation does



Hand-written prompt

"Answer the question based
only on the following
context: {context}
Question: {question}
Answer: "



Signature

vs.

"context, question --> answer"

Signature

Module

"context, question --> answer"



ChainOfThought



Automatically generated
prompt

"Given the fields `context`,
'question', produce the fields
'answer'.

Follow the following format.
Context: \${context}
Question: \${question}

Reasoning: Let's think step by
step in order to \${produce the
answer}. We ...
Answer: \${answer}"

Initial prompt

"Given the fields 'context', 'question', produce the fields 'answer'.

Follow the following format.

Context: \${context}

Question: \${question}

Reasoning: Let's think step by step in order to \${produce the answer}. We ...

Answer: \${answer}"



Optimized prompt

"Answer questions with short factoid answers.

Given the fields 'context', 'question', produce the fields 'answer'.

Question: What was the first computer language the author learned?
Answer: Fortran

Question: What kind of writing did the author do before college?
Answer: Short stories

Follow the following format.

Context: \${context}

Question: \${question}

Reasoning: Let's think step by step in order to \${produce the answer}. We ...

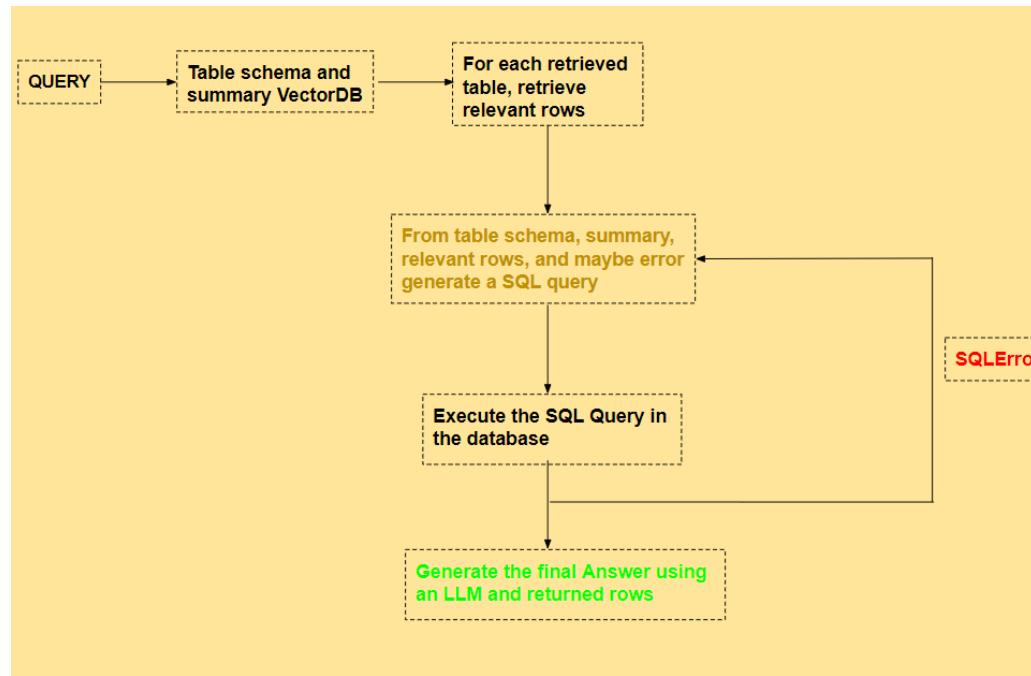
Answer: \${answer}"

DSPy For RAG example

```
1 class RAG(dspy.Module):
2     def __init__(self, num_passages=3):
3         # 'Retrieve' will use the user's default retrieval settings unless overriden.
4         self.retrieve = dspy.Retrieve(k=num_passages)
5         # 'ChainOfThought' with signature that generates answers given retrieval & question.
6         self.generate_answer = dspy.ChainOfThought("context, question -> answer")
7
8     def forward(self, question):
9         context = self.retrieve(question).passages
10        return self.generate_answer(context=context, question=question)
```

```
1 # Small training set with only questions and final answers.
2 qa_trainset = [dspy.Example(question="What is the capital of France?", answer="Paris")]
3
4 # The teleprompter will bootstrap missing labels: reasoning chains and retrieval contexts.
5 teleprompter = dspy.BootstrapFewShot(metric=dspy.evaluate.answer_exact_match)
6 compiled_rag = teleprompter.compile(RAG(), trainset=qa_trainset)
```

DSPy framework for Text to SQL questions



Text to SQL generation practical challenges

- Context Size Limitation
- Reaching Token Limit
- Determining the Best Prompt
- Model Hallucination
- Evaluating Performance
- Schema vs. Dataset Inclusion
- Handling Industrial-Scale Data

Take-home message

- Prompt engineering is powerful, particularly combined with in-context learning and Chain of Thought reasoning
- Prompt engineering is foundational for applications like AI agent development
- Prompt engineering and LLMs can provide effective solutions for Text to SQL generation tasks, especially along with the sub tasks SQL optimization and SQL debugging, schema link learning

Prompt engineering live demo

References

- Prompt Engineering Strategies: <https://platform.openai.com/docs/guides/prompt-engineering>
- VannaAI: <https://github.com/vanna-ai/vanna/blob/main/papers/ai-sql-accuracy-2023-08-17.md>
- Citation: Zhang B, Ye Y, Du G, Hu X, Li Z, Yang S, Liu CH, Zhao R, Li Z, Mao H. Benchmarking the Text-to-SQL Capability of Large Language Models: A Comprehensive Evaluation. arXiv preprint arXiv:2403.02951. 2024 Mar 5. [[pdf](#)]
- LangChain: https://python.langchain.com/v0.1/docs/use_cases/sql/quickstart/
- DSPy: <https://github.com/stanfordnlp/dspy?tab=readme-ov-file>
- Text2SQL: The Dream versus Reality: <https://www.youtube.com/watch?v=xmzda44hUgk>

Q&A