6000E Project Presentation

Group 4

Jiajun Zhao, Yuan Chen. (in alphabetical order)

Classic Project: Text to SQL

April 26, 2024

What is the problem?

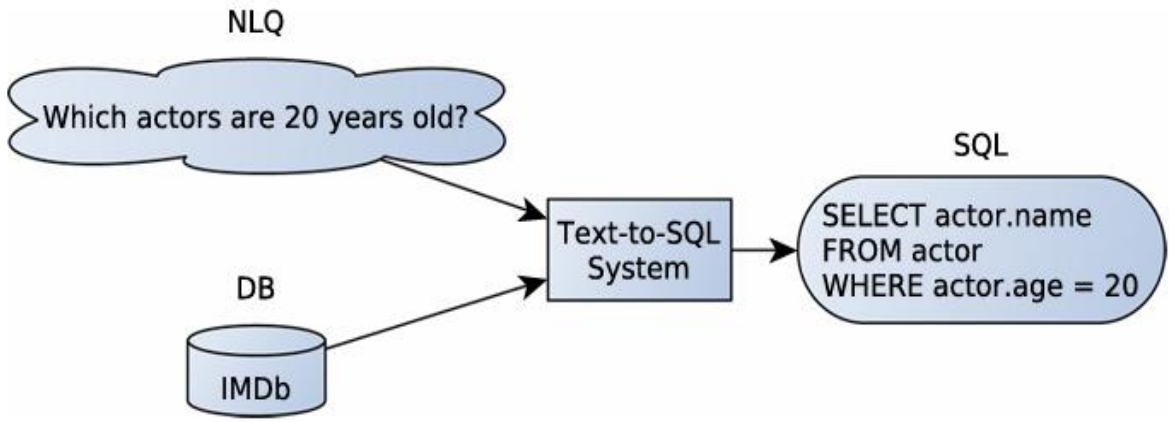


Fig. 1 The text-to-SQL problem

Definition:
Text-to-SQL (or Text2SQL), as the name implies, is to convert text into SQL.

Purpose:
let non-expert users to look-up database easily.

https://link.springer.com/article/10.1007/s00778-022-00776-8

Dataset and Data Processing

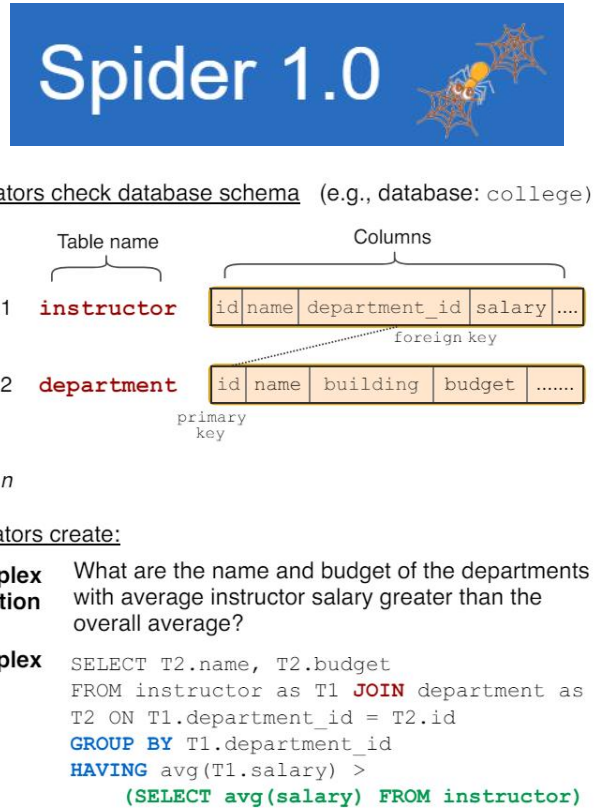


Figure 1: Our corpus annotates complex questions and SQLs. The example contains joining of multiple tables, a GROUP BY component, and a nested query.

For each question, generate a training sample based on the description of the question and the associated database description. This sample includes the question, the corresponding SQL query statement, and the database description associated with the question.


```
{
    "db_id": "pets_1",
    "instruction": "I want you to act as a SQL terminal in front of an example database, you need only to return the sql command to me.Below is an instruction that describes a task, Write a response that appropriately completes the request.\n\"\n##Instruction:\npets_1 contains tables such as Student, Has_Pet, Pets. Table Student has columns such as StuID, LName, Fname, Age, Sex, Major, Advisor, city_code. StuID is the primary key.\nTable Has_Pet has columns such as StuID, PetID. Table Pets has columns such as PetID, PetType, pet_age, weight. PetID is the primary key.\nThe StuID of Has_Pet is the foreign key of StuID of Student.\nThe PetID of Has_Pet is the foreign key of PetID of Pets.\n\n",
    "input": "###Input:\nFind the type and weight of the youngest pet.\n\n###Response:",
    "output": "SELECT pettype ,  weight FROM pets ORDER BY pet_age LIMIT 1",
    "history": []
},
```


https://github.com/eosphoros-ai/DB-GPT-Hub/blob/main/dbgpt_hub/data_process/sql_data_process.py

Fine tuning with lora and llamapro

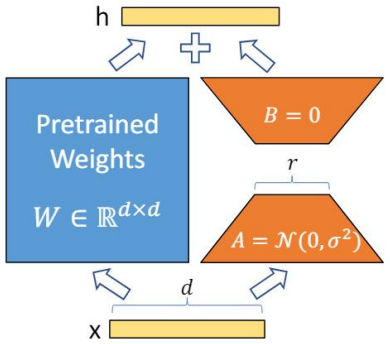
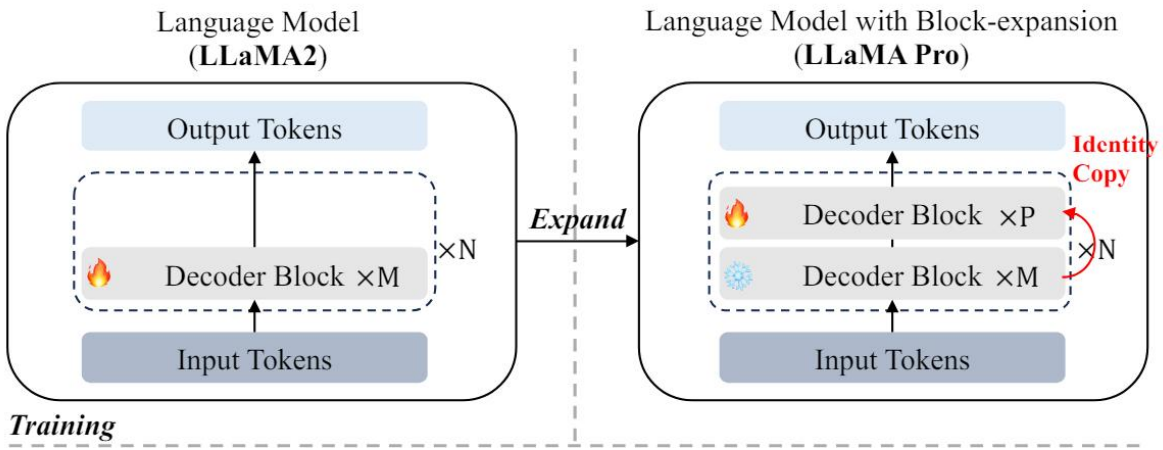


Figure 1: Our reparametrization. We only train A and B.

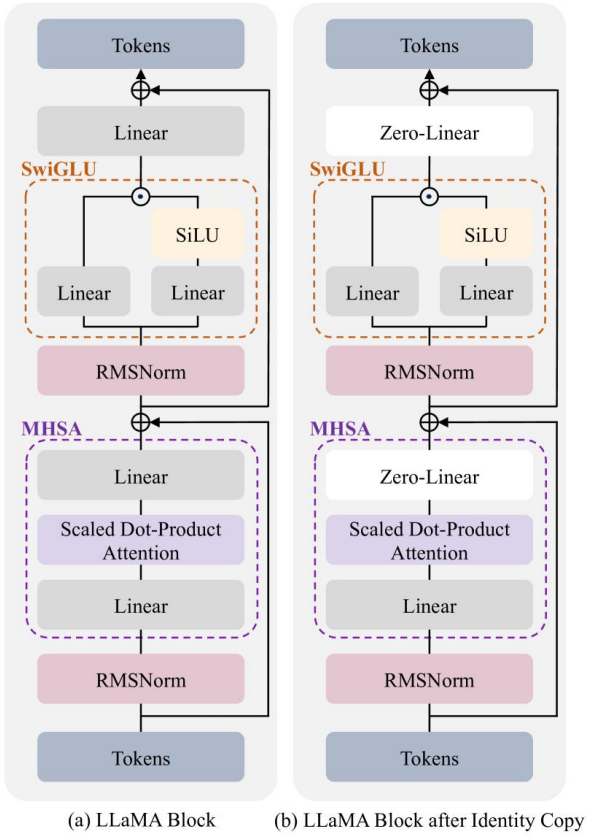
LoRA

trainable params%: 0.0472



LLaMA Pro

trainable params%: 12.0135
(4blocks/32Transformer Decoder layer)



(a) LLaMA Block (b) LLaMA Block after Identity Copy

LLaMA-Factory
Easy and Efficient LLM Fine-Tuning

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY (GUANGZHOU)

Evaluation on Spider

Execution Accuracy (EX): Calculate the proportion of the correct number of SQL execution results in the data set.
Exact Match (EM): Calculate the matching degree between the SQL generated by the model and the marked SQL.

model	finetune type	Execution Accuracy (EX)	Exact Match (EM)
sqlcoder2 7b	none	0.489	0.053
sqlcoder2 7b	lora	0.574	0.269
sqlcoder2 7b	llamapro	0.741	0.646
codellama 7b	lora	0.702	0.521
codellama 7b	llamapro	0.750	0.571
codeqwen1.5 7b	none	0.689	0.446
codeqwen1.5 7b	lora	0.767	0.708
codeqwen1.5 7b	llamapro	**0.798**	**0.708**

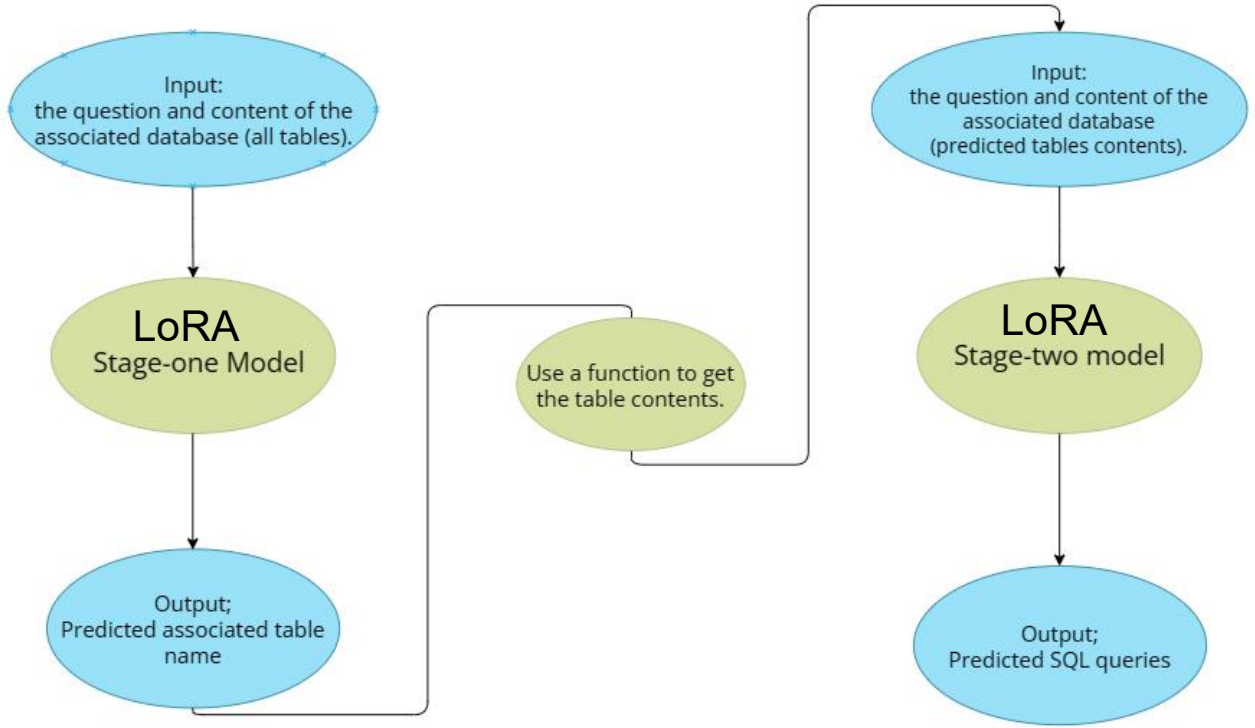
https://github.com/taoyds/test-suite-sql-eval/tree/master?tab=readme-ov-file

Pipeline: DTS-SQL

Table Identification Error

medium pred: SELECT feature_type_name FROM ref_feature_types WHERE feature_type_code = "AirCon";
medium gold: SELECT T2.feature_type_name FROM Other_Available_Features AS T1 JOIN Ref_Feature_Types AS T2 ON T1.feature_type_code = T2.feature_type_code WHERE T1.feature_name = "AirCon"

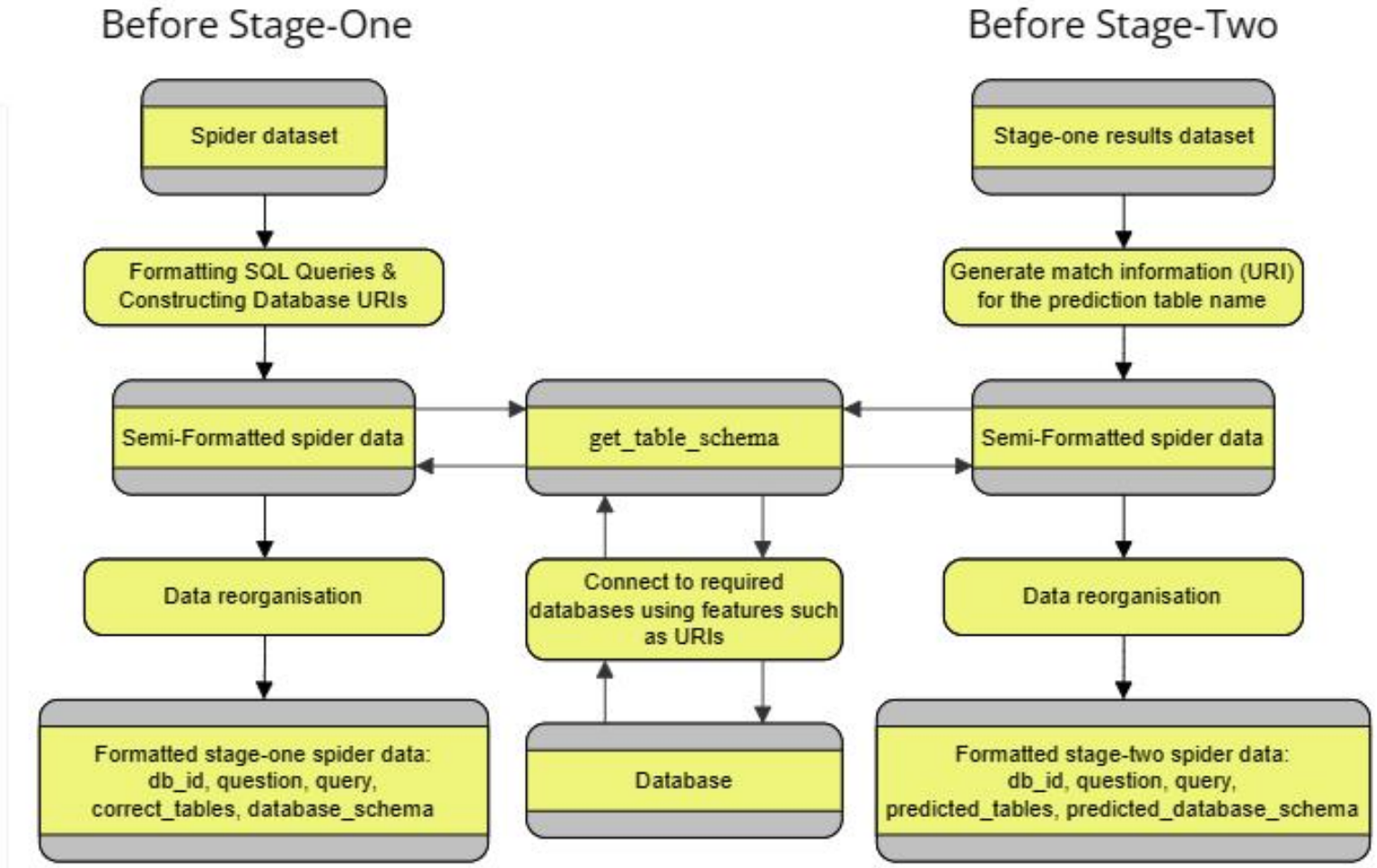
DTS-SQL is a two-stages fine-tuning pipeline.



https://arxiv.org/abs/2402.01117

Data Prepocessing

Spider 1.0



Evaluation

Comparison with our Single-stage fine-tuning result:

model	way	Execution Accuracy (EX)	Exact Match (EM)
codeqwen1.5 7b	Single-stage	0.798	0.708
codeqwen1.5 7b	Two-stage	0.813 ⬆	0.736 ⬆

Comparison with other opensource model result:

EM EX

3	92.5 (2020/11-IE-SQL+Execution-Guided Decoding)	73.9 (2022/09-CatSQL + GraPPa)	86.2 (2023/08-DAIL-SQL + GPT-4)	68.90 (2024/02-PB-SQL)	64.84 (2024/02-PB-SQL v1)
4	92.2 (2020/03-HydraNet+Execution-Guided Decoding)	73.1 (2022/09-SHiP + PICARD)	85.6 (2023/10-DPG-SQL + GPT-4 + Self-Correction)	68.80 (2024/04-OpenSearch-SQL,v1 + GPT-4)	63.39 (2024/02-SENSE 13B)
5	91.9 (2020/12-BRIDGE+Execution-Guided Decoding)	72.9 (2022/05-G^{3}R + LGESQL + ELECTRA)	85.3 (2023/04-DIN-SQL + GPT-4)	67.68 (2023/11-MAC-SQL + GPT-4)	63.22 (2024/04-GRA-SQL)
6	91.8 (2019/08-X-SQL+Execution-Guided Decoding)	72.4 (2022/08-RESDSQL+T5-1.1-lm100k-xl)	83.9 (2023/07-Hindsight Chain of Thought with GPT-4)	64.52 (2024/02-DTS-SQL + DeepSeek 7B)	60.98 (2024/03-Chat2Query)
7	91.4 (2021/03-SDSQL)	72.4 (2022/05-T5-SR)	82.3 (2023/06-C3 + ChatGPT + Zero-Shot)	64.22 (2023/10-SFT CodeS-15B)	60.71 (2023/11-Dubo-SQL-v1)

Model	Tuning	EX	EM
Mistral 7B	FT Tuning	67.0	63.9
Mistral 7B	DTS-SQL	71.1	64.6
Mistral 7B	Upper bound	81.9	74.5
DeepSeek 7B	FT Tuning	70.4	56.6
DeepSeek 7B	DTS-SQL	76.2	68.9
DeepSeek 7B	Upper bound	85.5	78.1

Table 6: Performance of the LLMs with different tuning methods on Spider-SYN dev set. FT stands for Full tables finetuning, Upper bound performance is the performance which we can achieve with a perfect schema linking.

Inference

vLLM

inference	Execution Accuracy	Exact Match	time
transformer	0.813	0.736	4h15min
vllm	0.686	0.610	277s

Faster but worse performence

inference	EX	EM	time
2stage transformer	0.813	0.736	4h15min
1st transformer 2st vllm	0.787	0.703	around 2h
1st vllm 2st transformer	0.695	0.627	around 2h
2stages vllm	0.686	0.610	277s

Analysis

1. **Reason for LLaMAPro's Better Performance:** One possible reason why LLaMAPro performs better than LoRA could be due to the increased number of model parameters after fine-tuning, which is around 8 billion. Another is the observed higher accuracy of LLaMAPro's Table Identification compared to LoRA, which could also contribute to LLaMAPro's better performance.
2. **Potential for Improvement in Stage 1**: There is still room for improvement in stage 1. The upper bound scenario, where all tables are identified correctly, shows even better performance.
3. **Applicability of the 2-Stage Method**: The two-stage method may not be well-suited for standard production environments, as each model requires approximately 14GB of VRAM.

model	way	EX	EM
codeqwen	upperbound	0.876	0.809
codeqwen	model pred	0.813	0.736

upperbound:
Upper bound performance is the performance which we can achieve with a perfect schema linking.

6000E Project Presentation

Group 4

Thanks for listening.