

«*In The Name Of GOD*»



دانشگاه صنعتی امیر کبیر
(پلی تکنیک تهران)

[HW-08-Report]

[NEURAL COMPUTING AND DEEP LEARNING]

Hasan Masroor | [403131030] | July 27, 2025

"فهرست مطالب تمرین 08"

Question 1	2
1)	2
2)	4
3)	14
4)	17
5)	19
6)	19

Problem 8: SQL

Question 1

1.

در این تمرین، ما به طراحی و ارزیابی یک سامانه تبدیل جملات زبان طبیعی به کوئری‌های SQL با استفاده از مدل‌های زبانی بزرگ می‌پردازیم. هدف ما بررسی تأثیر پارامترهای مختلف مانند تنظیم دقیق و مهندسی پرامپت بر عملکرد مدل است. همچنین، با افزودن معماری تولید تقویت‌شده با بازیابی (RAG)، کارایی سیستم را ارتقاء داده و تأثیر آن را بر دقت بررسی خواهیم کرد. در نهایت، عملکرد کلی سامانه با معیارهای Exact Match و Jaccard Similarity سنجیده شده و تحلیل‌های لازم در خصوص چالش‌ها و راهکارهای موجود ارائه می‌شود.

در این بخش، هدف اصلی ما طراحی و ارزیابی یک مدل پایه برای تبدیل پرسش‌های زبان طبیعی به کوئری‌های SQL بود. برای دستیابی به این هدف، ما از مدل زبانی بزرگ Gemma-2b-it استفاده کردیم که از نسخه‌های کوچک‌تر مدل‌های Gemma است و از طریق کتابخانه Hugging Face Transformers بارگذاری شد. انتخاب این مدل به دلیل توانایی‌های آن در تولید متن و سازگاری با محیط‌های محاسباتی دارای منابع محدود صورت گرفت. برای بهینه‌سازی مصرف حافظه و افزایش کارایی، مدل با استفاده از کوانتیزاسیون ۴-بیتی بارگذاری شد؛ این تکنیک با کاهش دقت عددی پارامترهای مدل، امکان اجرای آن را بر روی سخت‌افزارهایی با حافظه کمتر فراهم می‌کند، بدون اینکه افت عملکرد قابل توجهی داشته باشد. پس از بارگذاری مدل و توکنایزر، داده‌های مربوط به مجموعه Spider بارگذاری شدند؛ این مجموعه داده شامل جملات زبان طبیعی به همراه کوئری‌های متناظر SQL و ساختار پایگاه داده مربوط به هر پرسش است. فایل‌های train_spider.json، train_others.json، train_gold.sql، dev.json و tables.json از این مجموعه برای آموزش و ارزیابی استفاده شدند. توابع کمکی برای بارگذاری فایل‌های JSON و SQL و همچنین استخراج و فرمت‌بندی اسکیمای پایگاه داده به شکلی قابل فهم برای مدل، پیاده‌سازی شدند.

مهم‌ترین گام در این مرحله، ساخت پرامپت برای هر پرسش بود. پرامپت شامل دستورالعمل‌های کلی برای مدل، اسکیمای پایگاه داده مربوطه و پرسش زبان طبیعی بود. این ساختاردهی به مدل کمک می‌کرد تا با استفاده از اطلاعات شماتیک، کوئری SQL صحیح را تولید کند. در فاز ارزیابی، مدل بر روی داده‌های dev.json اجرا شد. برای هر نمونه، پرامپت متناظر ساخته و سپس مدل با استفاده از متد model.generate و تنظیماتی دیگر کوئری SQL را تولید کرد. خروجی مدل پردازش شد تا تنها کوئری SQL نهایی استخراج و پاکسازی شود. در نهایت، عملکرد مدل با استفاده از دو معیار کلیدی دقت تطابق کامل (Exact Match Accuracy) و امتیاز شباهت ژاکارد (Jaccard Similarity Score) ارزیابی شد. معیار Exact Match برابری دقیق کوئری تولیدی با کوئری مرجع را بررسی می‌کند، در حالی که Jaccard Similarity شباهت مجموعه‌ای از توکن‌های دو کوئری را اندازه‌گیری می‌کند. همچنین میانگین این معیارها بر روی تمام نمونه‌های پردازش شده، به عنوان شاخصی از عملکرد مدل پایه گزارش شد. در ادامه تعداد نمونه‌های مجموعه داده را نمایش دادیم:

```
Model loaded successfully.
Loaded 8659 training samples.
Loaded 2047 gold SQL queries.
Loaded 1034 dev samples.
Loaded 166 table schemas.
```

در ادامه، جهت درک بهتر عملکرد مدل پایه، تعدادی از نمونه خروجی‌های تولید شده توسط مدل به همراه پرسش‌های زبان طبیعی و کوئری‌های SQL مرجع (Gold SQL) ارائه شده‌اند. این نمونه‌ها، هم موارد موفقیت‌آمیز و هم مواردی که مدل در تولید کوئری دقیق دچار چالش شده است را نشان می‌دهند. برای هر نمونه، مقادیر EM و JS نیز برای ارزیابی کیفی ارائه گردیده‌اند:

```
--- Sample 1 ---
Database ID: concert_singer
Question: How many singers do we have?
Generated SQL: SELECT COUNT(*) FROM singer
Gold SQL: SELECT count(*) FROM singer
Exact Match (EM): 1.0000, Jaccard Score (JS): 1.0000

--- Sample 2 ---
Database ID: concert_singer
Question: What is the total number of singers?
Generated SQL: SELECT COUNT(*) FROM singer
Gold SQL: SELECT count(*) FROM singer
Exact Match (EM): 1.0000, Jaccard Score (JS): 1.0000

--- Sample 3 ---
Database ID: concert_singer
Question: Show name, country, age for all singers ordered by age from the oldest to the youngest.
Generated SQL: SELECT s.Name, s.Country, s.Age
FROM singer s
ORDER BY s.Age DESC
Gold SQL: SELECT name , country , age FROM singer ORDER BY age DESC
Exact Match (EM): 0.0000, Jaccard Score (JS): 0.4286
```

```
Final Evaluation Results for Base Model:
Average Exact Match (EM): 0.0484
Average Jaccard Similarity (JS): 0.3854
```

نتایج ارزیابی مدل پایه نشان می‌دهد که میانگین دقت تطابق کامل (Exact Match) برابر با ۰,۰۴۸۴ و میانگین امتیاز شباهت ژاکارد (Jaccard Similarity) برابر با ۰,۳۸۵۴ است. این مقادیر اولیه، در نگاه اول، ممکن است پایین به نظر برسند، اما باید توجه داشت که این نتایج مربوط به مدل پایه و بدون هیچ‌گونه بهینه‌سازی یا تنظیم دقیق خاصی است. مدل در این مرحله تنها با استفاده از ساختار پرامپت و اطلاعات اسکیمای پایگاه داده، تلاش به تولید کوئری‌های SQL کرده است. با این حال، نمره Jaccard Similarity حدود ۰,۳۸۵۴ نشان‌دهنده این

است که مدل تا حدودی قادر به شناسایی و استفاده از توکن‌های صحیح (مانند نام جداول و ستون‌ها، کلمات کلیدی SQL) در کوئری‌های تولیدی بوده است. تفاوت قابل توجه بین Exact Match و Jaccard Similarity نیز نشان دهنده این نکته است که اگرچه مدل در تولید کوئری‌های کاملاً منطبق با مرجع (از نظر ترتیب و ساختار دقیق) چالش دارد، اما در تشخیص اجزای اصلی کوئری موفقیت‌هایی کسب کرده است.

2.

در ادامه، با هدف درک عمیق‌تر از رفتار و عملکرد مدل در شرایط مختلف، به بررسی تأثیر پارامترهای گوناگون بر دقت تبدیل زبان طبیعی به SQL خواهیم پرداخت. این بخش شامل تحلیل تأثیر تنظیم دقیق، تعداد گام‌های گرم کردن و دمای تولید است.

▪ Fine-tuning

برای ارتقاء چشمگیر عملکرد مدل پایه، رویکرد تنظیم دقیق (Fine-Tuning) را با استفاده از تکنیک QLoRA (Quantized Low-Rank Adaptation) در پیش گرفتیم. این روش به ما امکان می‌دهد که مدل‌های زبانی بزرگ را با مصرف منابع محاسباتی به مراتب کمتر، به طور موثر تنظیم کنیم. در این پیاده‌سازی، مدل Gemma-2b-it مجدداً با کوانتیزاسیون ۴-بیتی بارگذاری شد تا حافظه مورد نیاز به حداقل برسد. پس از آماده‌سازی مدل برای آموزش با کوانتیزاسیون، یک پیکربندی LoRA تعریف شد. LoRA با افزودن ماتریس‌های کوچک و قابل آموزش به لایه‌های خاصی از مدل (مانند `q_proj`, `v_proj`, `k_proj`, `gate_proj`, `up_proj`, `down_proj`)، به ما اجازه می‌دهد تنها بخش کوچکی از پارامترهای مدل را آموزش دهیم و در عین حال، عملکرد مدل را به طور قابل توجهی بهبود بخشیم. این رویکرد به ویژه در سناریوهایی که داده‌های آموزشی محدود هستند و نیاز به حفظ دانش پیشین مدل وجود دارد، کارایی بالایی دارد. با این تنظیمات، تنها درصد کمی از کل پارامترهای مدل، قابل آموزش شدند که سرعت و کارایی فرآیند تنظیم دقیق را افزایش داد. برای آموزش مدل، داده‌های آموزشی مجموعه Spider شامل `train_spider.json` و `train_others.json` بارگذاری و برای ایجاد پرامپت‌های مناسب، با اسکیمای پایگاه داده ترکیب شدند. هر نمونه آموزشی شامل پرامپت و کوئری SQL مرجع متناظر با آن بود. این پرامپت‌های کامل، توکنایز شده و برای آموزش مدل آماده شدند. جهت تسریع در فرآیند تنظیم دقیق، زیرمجموعه‌ای از داده‌ها (۳۰٪) به صورت تصادفی انتخاب و مورد استفاده قرار گرفتند.

فرآیند آموزش با استفاده از Trainer در کتابخانه Transformers و با تنظیمات مشخصی آغاز شد. این تنظیمات شامل تعداد اپوک‌ها، اندازه بچ، گام‌های انباشت گرادین، نرخ یادگیری و گام‌های گرم کردن بود که به منظور بهینه‌سازی فرآیند آموزش و جلوگیری از `overfitting` تعیین شدند. پس از اتمام آموزش، مدل تنظیم شده و توکنایزر آن ذخیره شدند تا در مراحل بعدی برای ارزیابی و مقایسه با مدل پایه مورد استفاده قرار گیرند. این گام انتظار می‌رود که دقت مدل در تبدیل زبان طبیعی به SQL را به طور چشمگیری افزایش دهد.

Step	Training Loss		
10	2.505300		
20	1.580200		
30	0.989800	260	0.087700
40	0.734500	270	0.066500
50	0.534700	280	0.073200
60	0.439400	290	0.078600
70	0.332700	300	0.072400
80	0.252700	310	0.078400
90	0.225800	320	0.071300
100	0.168000	330	0.071500
110	0.150500	340	0.061100
120	0.141300	350	0.062300
130	0.121100	360	0.057100
140	0.136000	370	0.053400
150	0.115600	380	0.059500
160	0.112900	390	0.055100
170	0.091800	400	0.050700
180	0.101100	410	0.058300
190	0.090300	420	0.052700
200	0.077500	430	0.057600
210	0.088900	440	0.056700
220	0.077500	450	0.058400
230	0.085100	460	0.055200
240	0.093200	470	0.055000
250	0.079900	480	0.047300

این جدول، روند کاهش خطای آموزش را طی فرآیند فاین تیونینگ نشان می‌دهد. همانطور که مشاهده می‌شود، در گام‌های اولیه، کاهش قابل توجهی در خطا به چشم می‌خورد که حاکی از یادگیری سریع مدل است. پس از گام‌های نخستین، شیب کاهش خطای آموزش ملایم‌تر می‌گردد، اما روند نزولی همچنان پایدار می‌ماند. این پدیده نشان‌دهنده بهبود مستمر عملکرد مدل و تسلط فزاینده آن بر داده‌های آموزشی است. دستیابی به مقادیر پایین خطا در مراحل پایانی، گواهی بر موفقیت این مرحله از آموزش است و این انتظار را تقویت می‌کند که مدل تنظیم‌شده، عملکردی به مراتب بهتر از مدل پایه ارائه دهد.

در ادامه نیز تعدادی از خروجی‌ها را مثل قبل نمایش دادیم:

```

--- Sample 1 ---
Database ID: concert_singer
Question: How many singers do we have?
Generated SQL: SELECT count(*) FROM singer
Gold SQL: SELECT count(*) FROM singer
Exact Match (EM): 1.0000, Jaccard Score (JS): 1.0000

--- Sample 2 ---
Database ID: concert_singer
Question: What is the total number of singers?
Generated SQL: SELECT count(*) FROM singer
Gold SQL: SELECT count(*) FROM singer
Exact Match (EM): 1.0000, Jaccard Score (JS): 1.0000

--- Sample 3 ---
Database ID: concert_singer
Question: Show name, country, age for all singers ordered by age from the oldest to the youngest.
Generated SQL: SELECT name , country , age FROM singer ORDER BY age DESC
Gold SQL: SELECT name , country , age FROM singer ORDER BY age DESC
Exact Match (EM): 1.0000, Jaccard Score (JS): 1.0000

--- Sample 4 ---
Database ID: concert_singer
Question: What are the names, countries, and ages for every singer in descending order of age?
Generated SQL: SELECT name , country , age FROM singer ORDER BY age DESC
Gold SQL: SELECT name , country , age FROM singer ORDER BY age DESC
Exact Match (EM): 1.0000, Jaccard Score (JS): 1.0000

```

پس از انجام فرآیند تنظیم دقیق مدل بر روی ۱۰۳۴ نمونه مورد ارزیابی قرار گرفت. نتایج این ارزیابی پیشرفت قابل ملاحظه‌ای را نسبت به مدل پایه نشان می‌دهد. میانگین دقت تطابق کامل به ۰٫۱۸۳۸ و میانگین امتیاز شباهت ژاکارد به ۰٫۶۰۹۶ افزایش یافته است. این بهبود چشمگیر در هر دو معیار، به ویژه افزایش سه برابری در دقت تطابق کامل و رشد قابل توجه در امتیاز ژاکارد، مؤید اثربخشی روش تنظیم دقیق با QLoRA در ارتقاء توانایی مدل Gemma-2b-it برای تبدیل دقیق‌تر جملات زبان طبیعی به کوئری‌های SQL است. این نتایج نشان می‌دهد که مدل پس از فاین‌تیونینگ، نه تنها اجزای صحیح کوئری را بهتر شناسایی می‌کند، بلکه ساختار و ترتیب آن‌ها را نیز با دقت بیشتری تولید می‌کند.

```

Evaluation Results (Fine-tuned Model):
Samples evaluated: 1034
Average Exact Match (EM): 0.1838
Average Jaccard Similarity (JS): 0.6096

```

- Warmup step = 50

Step	Training Loss		
10	2.383800	260	0.084600
20	1.238900	270	0.063700
30	0.810800	280	0.069900
40	0.578800	290	0.074500
50	0.384600	300	0.070100
60	0.317900	310	0.074900
70	0.238800	320	0.068300
80	0.193000	330	0.069200
90	0.186600	340	0.058700
100	0.137900	350	0.059700
110	0.126800	360	0.055000
120	0.117700	370	0.052400
130	0.104000	380	0.057600
140	0.117900	390	0.053000
150	0.105700	400	0.049300
160	0.101400	410	0.056200
170	0.086200	420	0.051400
180	0.091200	430	0.055600
190	0.081300	440	0.056100
200	0.069600	450	0.056200
210	0.082800	460	0.053100
220	0.075600	470	0.053300
230	0.081000	480	0.046500
240	0.087300		
250	0.075600		

در این بخش، به بررسی تأثیر پارامتر گام‌های گرم کردن در فرآیند تنظیم دقیق مدل پرداختیم. گام‌های گرم کردن، به تعداد گام‌های اولیه آموزش اشاره دارد که در آن‌ها نرخ یادگیری به تدریج از مقدار صفر یا یک مقدار کوچک به نرخ یادگیری کامل (تعیین‌شده) افزایش می‌یابد. این رویکرد به مدل کمک می‌کند تا در فاز ابتدایی آموزش، با تنظیمات نرخ یادگیری ملایم‌تر شروع کند و از ناپایداری‌های احتمالی در اوایل آموزش جلوگیری شود. در این آزمایش، مدل Gemma-2b-it با استفاده از QLoRA و با همان تنظیمات کلی فاین‌تیونینگ قبلی، اما با تنظیم warmup_steps بر روی مقدار ۵۰، آموزش داده شد. این تغییر در مقایسه با مقدار پیش‌فرض ۱۰۰ در آزمایش

فاین تیونینگ اولیه انجام گرفت. هدف از این آزمایش، بررسی حساسیت مدل به مدت زمان فاز گرم کردن و چگونگی تأثیر آن بر روند همگرایی و عملکرد نهایی مدل در وظیفه تبدیل زبان طبیعی به SQL بود. پس از اتمام آموزش با این تنظیم جدید، نتایج ارزیابی برای مقایسه با سایر سناریوها آماده خواهد شد.

در جدول صفحه قبل نیز روند کاهش خطای آموزش مدل را طی فرآیند فاین تیونینگ با ۵۰ گام گرم کردن نشان می دهد. مشابه آزمایش قبلی، در گام های اولیه شاهد کاهش سریع خطا هستیم که حاکی از یادگیری اولیه مدل است. پس از آن، روند کاهش ملایم تر شده و به سمت همگرایی پیش می رود. رسیدن به مقادیر پایین خطا در گام های پایانی، نشان دهنده موفقیت فرآیند آموزش و توانایی مدل در فیت شدن بر روی داده های آموزشی با این تنظیمات است. در ادامه تعدادی از نمونه های خروجی را نیز مشاهده می کنیم:

```
--- Sample 11 ---
Database ID: concert_singer
Question: Show all countries and the number of singers in each country.
Generated SQL: SELECT country , count(*) FROM singer GROUP BY country ORDER BY count(*) DESC LIMIT 10
Gold SQL: SELECT country , count(*) FROM singer GROUP BY country
Exact Match (EM): 0.0000, Jaccard Score (JS): 0.6667

--- Sample 12 ---
Database ID: concert_singer
Question: How many singers are from each country?
Generated SQL: SELECT count(*) , country FROM singer GROUP BY country ORDER BY count(*) DESC LIMIT 10
Gold SQL: SELECT country , count(*) FROM singer GROUP BY country
Exact Match (EM): 0.0000, Jaccard Score (JS): 0.6667

--- Sample 13 ---
Database ID: concert_singer
Question: List all song names by singers above the average age.
Generated SQL: SELECT Song_Name FROM singer WHERE Age > (SELECT avg(age) FROM singer) ORDER BY song_name
Gold SQL: SELECT song_name FROM singer WHERE age > (SELECT avg(age) FROM singer)
Exact Match (EM): 0.0000, Jaccard Score (JS): 0.8333
```

Evaluation Results (Fine-tuned Model):
Samples evaluated: 1034
Average Exact Match (EM): 0.1702
Average Jaccard Similarity (JS): 0.6046

ارزیابی مدل تنظیم شده با ۵۰ گام گرم کردن، بر روی ۱۰۳۴ نمونه، میانگین دقت تطابق کامل را ۰٫۱۷۰۲ و میانگین امتیاز شباهت ژاکارد را ۰٫۶۰۴۶ نشان می دهد. با مقایسه این نتایج با عملکرد مدل تنظیم شده اولیه (با ۱۰۰ گام گرم کردن، EM: 0.1838 و JS: 0.6096)، مشاهده می شود که کاهش جزئی در هر دو معیار دقت رخ داده است. این امر نشان می دهد که کاهش تعداد گام های گرم کردن از ۱۰۰ به ۵۰، تأثیری منفی بر عملکرد نهایی مدل داشته است. این نتیجه می تواند حاکی از آن باشد که فاز گرم کردن کافی در ابتدای آموزش برای همگرایی بهینه مدل و دستیابی به عملکرد بالاتر، اهمیت دارد و کاهش آن ممکن است باعث عدم پایداری اولیه در آموزش و در نتیجه کاهش جزئی دقت نهایی شود. این تحلیل، بر اهمیت تنظیم دقیق پارامترهای آموزش مانند گام های گرم کردن در فرآیند فاین تیونینگ تأکید می کند.

- Warmup step = 75

Step	Training Loss		
10	1.203400	260	0.079300
20	0.554700	270	0.061800
30	0.435600	280	0.066500
40	0.306900	290	0.072400
50	0.228300	300	0.066700
60	0.195800	310	0.072400
70	0.161100	320	0.066500
80	0.136600	330	0.066500
90	0.141800	340	0.056500
100	0.115100	350	0.057800
110	0.114000	360	0.052700
120	0.102400	370	0.050100
130	0.093400	380	0.055000
140	0.106200	390	0.052600
150	0.097800	400	0.048100
160	0.096800	410	0.054500
170	0.081900	420	0.050700
180	0.083600	430	0.053900
190	0.077300	440	0.054300
200	0.066900	450	0.054400
210	0.077700	460	0.050800
220	0.069600	470	0.051300
230	0.078000	480	0.045000
240	0.079800		
250	0.071600		

در ادامه بررسی تأثیر گام‌های گرم کردن بر فرآیند تنظیم دقیق، مدل Gemma-2b-it با استفاده از QLoRA و با تنظیم warmup_steps بر روی مقدار ۷۵ آموزش داده شد. این آزمایش با هدف مشاهده چگونگی تأثیر مدت

زمان متوسط‌تر فاز گرم کردن (بین ۵۰ و ۱۰۰ گام قبلی) بر روند همگرایی و عملکرد نهایی مدل در وظیفه تبدیل زبان طبیعی به SQL طراحی شده است. جدول بالا روند کاهش خطای آموزش مدل را طی فرآیند فاین‌تیونینگ با ۷۵ گام گرم کردن نشان می‌دهد. همانند آزمایش‌های قبلی، در گام‌های اولیه شاهد کاهش قابل توجهی در خطا هستیم که بیانگر یادگیری سریع مدل در فاز اولیه آموزش است. با ادامه فرآیند، شیب کاهش ملایم‌تر شده و خطا به تدریج به سمت مقادیر پایین‌تر همگرا می‌شود. رسیدن به خطای آموزش ۰,۰۴۵۰۰۰ در گام ۴۸۰، موفقیت این مرحله از آموزش را نشان می‌دهد و مؤید توانایی مدل در فیت شدن بر روی داده‌های آموزشی با این تنظیمات گام گرم کردن است.

برای درک بهتر مجدد تعدادی از نمونه‌های خروجی را در ادامه نمایش دادیم:

```

--- Sample S1 ---
Database ID: pets_1
Question: list the maximum weight and type for each type of pet.
Generated SQL: SELECT max(T1.weight) , T2.pettype FROM Pets AS T1 JOIN has_Pet AS T2 ON T1.petid = T2.petid GROUP BY T2.pettype ORDER BY sum(T1.weight) DESC LIMIT 1
Gold SQL: SELECT max(weight) , pettype FROM pets GROUP BY pettype
Exact Match (EM): 0.0000, Jaccard score (JS): 0.2500

--- Sample S2 ---
Database ID: pets_1
Question: Find number of pets owned by students who are older than 20.
Generated SQL: SELECT count(*) FROM has_Pet WHERE stuID IN (SELECT stuID FROM Student WHERE age > 20) GROUP BY stuID
Gold SQL: SELECT count(*) FROM student AS T1 JOIN has_pet AS T2 ON T1.stuid = T2.stuid WHERE T1.age > 20
Exact Match (EM): 0.0000, Jaccard Score (JS): 0.2917

--- Sample S3 ---
Database ID: pets_1
Question: How many pets are owned by students that have an age greater than 20?
Generated SQL: SELECT count(*) FROM has_Pet WHERE stuID IN (SELECT stuID FROM Student WHERE age > 20) GROUP BY stuID HAVING count(*) = 2
Gold SQL: SELECT count(*) FROM student AS T1 JOIN has_pet AS T2 ON T1.stuid = T2.stuid WHERE T1.age > 20
Exact Match (EM): 0.0000, Jaccard Score (JS): 0.1077

```

Evaluation Results (Fine-tuned Model):
Samples evaluated: 1034
Average Exact Match (EM): 0.1586
Average Jaccard Similarity (JS): 0.5838

ارزیابی مدل تنظیم‌شده با ۷۵ گام گرم کردن، بر روی ۱۰۳۴ نمونه، میانگین دقت تطابق کامل را ۰,۱۵۸۶ و میانگین امتیاز شباهت ژاکارد را ۰,۵۸۳۸ نشان می‌دهد. با مقایسه این نتایج با عملکرد مدل تنظیم‌شده اولیه (با ۱۰۰ گام گرم کردن، EM: 0.1838 و JS: 0.6096) و همچنین ۵۰ گام گرم کردن (EM: 0.1702 و JS: 0.6046)، مشاهده می‌شود که با کاهش تعداد گام‌های گرم کردن، افت تدریجی در هر دو معیار دقت تداوم یافته است. این روند، اهمیت وجود یک فاز گرم کردن کافی در ابتدای آموزش را برای دستیابی به همگرایی بهینه و عملکرد بالا در مدل تأیید می‌کند و نشان می‌دهد که مقادیر کمتر گام‌های گرم کردن، ممکن است از پایداری و دقت نهایی مدل بکاهد.

تعداد گام گرم کردن	50	75	100*
EM	0.1702	0.1586	0.1838
JS	0.6046	0.5838	0.6096

همانطور که در جدول بالا مشاهده می‌شود، نتایج ارزیابی مدل با تعداد گام‌های گرم کردن متفاوت، تأثیر این پارامتر حیاتی را بر عملکرد نهایی مدل تنظیم‌شده به وضوح نشان می‌دهد. بهترین عملکرد در هر دو معیار دقت تطابق کامل (۰,۱۸۳۸) و امتیاز شباهت ژاکارد (۰,۶۰۹۶)، زمانی حاصل شده است که تعداد گام‌های گرم کردن برابر با ۱۰۰ بوده است. با کاهش تعداد گام‌های گرم کردن به ۷۵، شاهد افت در هر دو معیار هستیم؛ به طوری که دقت تطابق کامل به ۰,۱۵۸۶ و امتیاز شباهت ژاکارد به ۰,۵۸۳۸ کاهش می‌یابد. این روند نزولی با کاهش بیشتر گام‌ها به ۵۰ نیز ادامه پیدا می‌کند، هرچند که در ۵۰ گام، نتایج (EM: 0.1702 و JS: 0.6046) کمی بهتر از ۷۵ گام است. این نتیجه غیرخطی، می‌تواند نشان‌دهنده پیچیدگی تأثیر این پارامتر باشد، اما به طور کلی، وجود یک فاز گرم کردن طولانی‌تر (مانند ۱۰۰ گام) برای همگرایی بهینه و دستیابی به بالاترین دقت، ضروری به نظر می‌رسد. این تحلیل تأکید می‌کند که تنظیم دقیق پارامترهای آموزش، از جمله تعداد گام‌های گرم کردن، نقش بسزایی در ارتقاء عملکرد مدل‌های زبانی در وظایف تخصصی مانند تبدیل زبان طبیعی به SQL دارد. انتخاب بهینه این پارامتر می‌تواند به مدل کمک کند تا در ابتدای فرآیند یادگیری به پایداری مناسبی دست یابد و از رسیدن به عملکردی زیر حد مطلوب جلوگیری کند.

▪ Temperature = 0.5

در این بخش، تأثیر پارامتر دمای تولید بر کیفیت کوئری‌های SQL تولیدی توسط مدل تنظیم‌شده بررسی شد. دمای تولید، عاملی است که میزان تصادفی بودن خروجی مدل را کنترل می‌کند. مدل تنظیم‌شده بارگذاری شد و تولید کوئری‌ها برای مجموعه داده dev.json با تنظیم دما روی ۰,۵ و فعال‌سازی نمونه‌برداری انجام گرفت. هدف از این تنظیم، ایجاد تعادلی بین دقت و تنوع در خروجی‌ها بود. عملکرد مدل پس از این تغییر، با معیارهای دقت تطابق کامل و امتیاز شباهت ژاکارد مجدداً ارزیابی شد تا تأثیر این پارامتر بر نتایج نهایی مشخص گردد.

در ادامه نمونه ای از خروجی این بخش را نشان داده ایم:

```

--- Sample 29 ---
Database ID: concert_hinger
Question: Show the stadium names without any concert.
Generated SQL: SELECT name FROM stadium EXCEPT SELECT T1.name FROM stadium AS T1 JOIN concert AS T2 ON T1.stadium_id = T2.stadium_id GROUP BY T1.name HAVING count(*) = 0
Gold SQL: SELECT name FROM stadium WHERE stadium_id NOT IN (SELECT stadium_id FROM concert)
Exact Match (EM): 0.0000, Jaccard Score (JS): 0.1538

--- Sample 30 ---
Database ID: concert_hinger
Question: What are the names of the stadiums without any concerts?
Generated SQL: SELECT name FROM stadium WHERE stadium_id NOT IN (SELECT stadium_id FROM concert)

### Output:
| name |
|---|---|
| Anfield Road |
| Molineux Stadium |
| Tottenham Hotspur Stadium |
| Tiverton Stadium |
| St. James' Park |
| Brighton & Hove Albion Stadium |
| Craven Cottage |
| Elland Road |
| King Power Stadium |
| Luton Town Stadium |
| Middlesbrough Stadium |
| Nottingham Forest Stadium |
Gold SQL: SELECT name FROM stadium WHERE stadium_id NOT IN (SELECT stadium_id FROM concert)
Exact Match (EM): 0.0000, Jaccard Score (JS): 0.2701

```

ارزیابی مدل تنظیم‌شده با دمای تولید ۰٫۵، بر روی ۱۰۳۴ نمونه، میانگین دقت تطابق کامل را ۰٫۱۷۳۱ و میانگین امتیاز شباهت ژاکارد را ۰٫۵۹۹۵ نشان داد. این مقادیر، کاهش جزئی را نسبت به نتایج قبلی مدل تنظیم‌شده (JS: 0.6096، EM: 0.1838) نشان می‌دهند. این افت اندک، حاکی از آن است که دمای ۰٫۵ برای این مدل و وظیفه خاص، نقطه بهینه نبوده و ممکن است کمی از دقت کاسته باشد.

```

Evaluation Results (Fine-tuned Model):
Samples evaluated: 1034
Average Exact Match (EM): 0.1731
Average Jaccard Similarity (JS): 0.5995

```

▪ Temperature = 1

در این آزمایش، به بررسی تأثیر افزایش دمای تولید به ۱ بر عملکرد مدل تنظیم‌شده پرداختیم. هدف این بود که چگونگی اثرگذاری دمای بالاتر بر تنوع خروجی‌های مدل و مقایسه آن با نتایج حاصل از دماهای قبلی مشخص شود. تعدادی از نمونه های خروجی را نیز مشاهده می‌کنیم:

```

--- Sample 87 ---
Database ID: pets_1
Question: what is the average age for all students who do not own any pets?
Generated SQL: SELECT avg(age) FROM student WHERE stuId NOT IN (SELECT stuId FROM Has_Pet) GROUP BY stuId
Gold SQL: SELECT avg(age) FROM student WHERE stuId NOT IN (SELECT T1.stuId FROM student AS T1 JOIN has_pet AS T2 ON T1.stuId = T2.stuId)
Exact Match (EM): 0.0000, Jaccard Score (JS): 0.4286

--- Sample 88 ---
Database ID: car_1
Question: How many continents are there?
Generated SQL: SELECT count(*) FROM continents
Gold SQL: SELECT count(*) FROM CONTINENTS;
Exact Match (EM): 0.0000, Jaccard Score (JS): 0.6000

```

ارزیابی مدل تنظیم‌شده با دمای تولید ۱، بر روی ۱۰۳۴ نمونه، میانگین دقت تطابق کامل را ۰,۱۶۴۴ و میانگین امتیاز شباهت ژاکارد را ۰,۵۸۶۶ نشان داد. با مقایسه این مقادیر با نتایج قبلی مدل تنظیم‌شده (EM: 0.1838، JS: 0.6096) و همچنین دمای ۰,۵ (JS: 0.5995، EM: 0.1731)، مشاهده می‌شود که با افزایش دما، کاهش تدریجی در هر دو معیار دقت تطابق کامل و شباهت ژاکارد رخ داده است. این روند نشان می‌دهد که افزایش دمای تولید به ۱، که منجر به افزایش تصادفی بودن در خروجی می‌شود، در این مورد خاص، به ضرر دقت مدل در تولید کوئری‌های SQL بوده است. این نتیجه تأکید می‌کند که برای وظایفی نظیر تبدیل زبان طبیعی به SQL که نیاز به دقت بالا و ساختارمندی دارند، مقادیر بالاتر دما ممکن است مناسب نباشند و نیاز به تعادلی دقیق‌تر بین خلاقیت و صحت وجود دارد.

Evaluation Results (Fine-tuned Model):
 Samples evaluated: 1034
 Average Exact Match (EM): 0.1644
 Average Jaccard Similarity (JS): 0.5866

دمای تولید	۰*	0.5	1
EM	0.1838	0.1731	0.1644
JS	0.6096	0.5995	0.5866

همانطور که در جدول بالا مشاهده می‌شود، تنظیم دقیق مدل با تکنیک QLoRA، تأثیر چشمگیری در بهبود عملکرد مدل داشته است. میانگین دقت تطابق کامل از ۰,۰۴۸۴ در مدل پایه به ۰,۱۸۳۸ در مدل تنظیم‌شده افزایش یافته که نشان‌دهنده ارتقاء قابل ملاحظه در توانایی مدل برای تولید کوئری‌های SQL دقیق است. امتیاز شباهت ژاکارد نیز از ۰,۳۸۵۴ به ۰,۶۰۹۶ رسیده که بیانگر بهبود در شناسایی و استفاده از توکن‌های صحیح در کوئری‌هاست. در خصوص تأثیر دمای تولید، نتایج نشان می‌دهد که افزایش این پارامتر از حالت پیش‌فرض (که عملاً نمونه‌برداری کنترل‌شده‌تری دارد) به ۰,۵ و سپس به ۱، منجر به کاهش تدریجی در هر دو معیار دقت تطابق کامل و شباهت ژاکارد شده است. این امر حاکی از آن است که برای وظیفه تبدیل زبان طبیعی به SQL، که نیاز به دقت و ساختارمندی بالایی دارد، افزایش تصادفی بودن خروجی توسط دماهای بالاتر، مطلوب نیست. به نظر می‌رسد مدل در شرایطی که فضای جستجوی کمتری برای تنوع دارد (یا دمای پایین‌تر است)، خروجی‌های دقیق‌تری تولید می‌کند. این تحلیل تأکید می‌کند که انتخاب دمای تولید مناسب برای وظایف حساس به دقت، از اهمیت بالایی برخوردار است و در این مورد، دماهای پایین‌تر به حفظ صحت ساختار کوئری‌ها کمک بیشتری کرده‌اند.

3.

در این بخش، تمرکز ما بر روی بررسی تأثیر مهندسی پرامپت بر دقت و کارایی مدل در تولید کوئری‌های SQL است. مهندسی پرامپت به معنای طراحی بهینه ورودی‌های متنی به مدل است تا خروجی‌های مطلوب‌تری حاصل شود. در این راستا، ما سه رویکرد اصلی شامل حالت بدون نمونه (Zero-shot)، حالت تک نمونه‌ای (One-shot) و حالت چند نمونه‌ای (Few-shot) را مورد آزمایش قرار خواهیم داد. هدف این است که با مقایسه عملکرد مدل در هر یک از این حالات، بهترین استراتژی برای ساخت پرامپت‌ها را شناسایی کنیم تا دقت نهایی تبدیل زبان طبیعی به SQL به حداکثر برسد.

▪ Zero-shot

در این رویکرد، برای ارزیابی عملکرد مدل بدون هیچ مثال قبلی در پرامپت، از همان تنظیمات و کد مدل پایه سوال ۱ استفاده شد. در این حالت، مدل صرفاً با دریافت پرسش زبان طبیعی و شمای پایگاه داده، بدون هیچ راهنمایی اضافی از طریق نمونه‌های ورودی-خروجی، به تولید کوئری SQL می‌پردازد. این رویکرد، توانایی مدل را در تعمیم دانش خود بدون وابستگی به مثال‌های درون پرامپت، مورد سنجش قرار می‌دهد.

```
Final Evaluation Results for Base Model:  
Average Exact Match (EM): 0.0484  
Average Jaccard Similarity (JS): 0.3854
```

همانطور که قبلاً نیز مشاهده شد، نتایج مدل پایه در حالت بدون نمونه (Zero-shot) نشان می‌دهد که میانگین دقت تطابق کامل برابر با ۰,۰۴۸۴ و میانگین امتیاز شباهت ژاکارد برابر با ۰,۳۸۵۴ است. این مقادیر، نقطه شروعی برای ارزیابی تأثیر مهندسی پرامپت محسوب می‌شوند و انتظار می‌رود با افزودن نمونه‌های راهنما، بهبود قابل توجهی در عملکرد مدل حاصل گردد.

▪ One-shot

در این رویکرد از مهندسی پرامپت، به منظور ارائه راهنمایی بیشتر به مدل، یک مثال واحد از زوج پرسش-پاسخ به پرامپت اضافه شد. این مثال، بلافاصله پیش از پرسش اصلی کاربر و پس از شمای پایگاه داده قرار می‌گیرد تا مدل با مشاهده یک الگوی صحیح، درک بهتری از فرمت مورد انتظار برای تولید کوئری SQL به دست آورد. این شیوه، با الگوبرداری از یک نمونه، به مدل کمک می‌کند تا پاسخ‌های مرتبط‌تر و دقیق‌تری تولید کند و پتانسیل بهبود عملکرد نسبت به حالت بدون نمونه را داراست.

```
Final Evaluation Results:  
Average Exact Match (EM): 0.0542  
Average Jaccard Similarity (JS): 0.3771  
  
Base model (One-Shot) evaluation complete.
```


ارزیابی مدل در حالت تک نمونه‌ای، میانگین دقت تطابق کامل را ۰,۰۵۴۲ و میانگین امتیاز شباهت ژاکارد را ۰,۳۷۷۱ نشان داد. با مقایسه این نتایج با حالت بدون نمونه (EM: 0.0484, JS: 0.3854)، مشاهده می‌شود که در دقت تطابق کامل افزایش جزئی وجود داشته، اما در امتیاز شباهت ژاکارد کاهش اندکی رخ داده است. این امر نشان می‌دهد که افزودن یک مثال واحد، در این مورد خاص، تأثیر چشمگیری بر بهبود کلی عملکرد مدل در تولید کوئری‌های دقیق نداشته است و برای هدایت بهتر مدل، شاید به نمونه‌های بیشتر یا مثال‌های هدفمندتری نیاز باشد.

▪ Few-shot

در این رویکرد پیشرفته‌تر از مهندسی پرامپت، به منظور ارائه راهنمایی جامع‌تر و دقیق‌تر به مدل، چندین مثال از زوج پرسش-پاسخ به پرامپت اضافه شد. این نمونه‌ها، مانند یک آموزش در لحظه برای مدل عمل می‌کنند و به آن کمک می‌کنند تا الگوهای پیچیده‌تر در تبدیل زبان طبیعی به SQL را تشخیص دهد. هر مثال شامل یک پرسش زبان طبیعی و کوئری SQL صحیح متناظر با آن است که مدل با مشاهده آن‌ها، قادر به تعمیم و تولید پاسخ‌های دقیق‌تر برای پرسش‌های جدید می‌شود. انتظار می‌رود این روش، به دلیل ارائه اطلاعات و راهنمایی بیشتر، منجر به بهبود قابل ملاحظه‌ای در عملکرد مدل نسبت به حالت‌های بدون نمونه و تک نمونه‌ای گردد.

```
Final Evaluation Results:
Average Exact Match (EM): 0.0638
Average Jaccard Similarity (JS): 0.4021
Base model (Few-shot) evaluation complete.
```

ارزیابی مدل در حالت چند نمونه‌ای، میانگین دقت تطابق کامل را ۰,۰۶۳۸ و میانگین امتیاز شباهت ژاکارد را ۰,۴۰۲۱ نشان داد. با مقایسه این نتایج با حالت‌های بدون نمونه و تک نمونه‌ای، شاهد بهبود قابل قبولی در هر دو معیار هستیم. این افزایش، مؤید اثربخشی ارائه چندین مثال راهنما به مدل در پرامپت است. این رویکرد به مدل کمک می‌کند تا الگوهای پیچیده‌تری را در تبدیل زبان طبیعی به SQL فرا گیرد و به تولید کوئری‌های دقیق‌تر نزدیک شود، که اهمیت مهندسی پرامپت را در ارتقاء عملکرد مدل‌های زبانی بزرگ برجسته می‌سازد.

Few-shot	One-shot	Zero-shot*	
0.0638	0.0542	0.0484	EM
0.4021	0.3771	0.3854	JS

همانطور که در جدول بالا مشاهده می‌شود، نتایج ارزیابی مدل در رویکردهای مختلف مهندسی پرامپت، نکات مهمی را ارائه می‌دهد. حالت بدون نمونه که صرفاً بر پایه دستورالعمل‌ها و شمای پایگاه داده عمل می‌کند،

پایین‌ترین دقت را در هر دو معیار نشان می‌دهد. در حالت تک نمونه‌ای، با افزودن یک مثال راهنما، شاهد افزایش جزئی در دقت تطابق کامل هستیم، اما امتیاز شباهت ژاکارد کمی کاهش یافته است. این موضوع نشان می‌دهد که یک مثال واحد همیشه برای بهبود پایدار و جامع کافی نیست. حالت چند نمونه‌ای بهترین عملکرد را در میان این سه رویکرد از خود نشان داده است. با ارائه چندین مثال، میانگین دقت تطابق کامل به ۰.۰۶۳۸ و میانگین امتیاز شباهت ژاکارد به ۰.۴۰۲۱ افزایش یافته است. این بهبود، اهمیت ارائه نمونه‌های کافی و مرتبط را در پرامپت برای هدایت مدل‌های زبانی بزرگ در وظایف پیچیده مانند تبدیل زبان طبیعی به SQL تأیید می‌کند. در نهایت، این تحلیل‌ها بر این نکته تأکید دارند که هرچند مدل پایه بدون تنظیم دقیق در این وظیفه چالش‌هایی دارد، اما مهندسی پرامپت، به ویژه در حالت چند نمونه‌ای، می‌تواند به طور مؤثری به بهبود عملکرد آن کمک کند.

▪ Impact of Schema Length on Performance

در این بخش، تأثیر حجم اطلاعات اسکیمای پایگاه داده در پرامپت بر عملکرد مدل مورد بررسی قرار گرفت. هدف این بود که مشخص شود آیا حذف یا خلاصه‌سازی جزئیات خاصی از ساختار پایگاه داده (مانند کلیدهای اصلی و خارجی) می‌تواند بر دقت مدل در تولید کوئری‌های SQL تأثیر مثبت یا منفی بگذارد. برای این منظور، تابع `get_db_schema_for_prompt` به گونه‌ای اصلاح شد که قابلیت کنترل بر درج اطلاعات کلیدهای اصلی و خارجی در شمای ورودی را داشته باشد. این آزمایش به ما کمک می‌کند تا درک بهتری از حساسیت مدل به میزان جزئیات ارائه‌شده در مورد ساختار پایگاه داده به دست آوریم.

```
Final Evaluation Results:  
Average Exact Match (EM): 0.0464  
Average Jaccard Similarity (JS): 0.3803
```

ارزیابی مدل با حذف اطلاعات کلیدهای اصلی و خارجی از شمای پایگاه داده در پرامپت، میانگین دقت تطابق کامل را ۰.۰۴۶۴ و میانگین امتیاز شباهت ژاکارد را ۰.۳۸۰۳ نشان داد. با مقایسه این نتایج با عملکرد مدل پایه در حالت بدون نمونه (EM: 0.0484، JS: 0.3854)، مشاهده می‌شود که کاهش جزئی در هر دو معیار رخ داده است. این امر حاکی از آن است که حذف این جزئیات، هرچند اندک، تأثیری منفی بر دقت مدل در تولید کوئری‌های SQL داشته است و نشان می‌دهد که حتی برای مدل پایه، وجود این اطلاعات می‌تواند در فهم دقیق‌تر روابط و ساختار پایگاه داده و در نتیجه تولید کوئری‌های صحیح‌تر، مؤثر باشد. این نتیجه بر اهمیت حفظ جزئیات کامل شمای پایگاه داده در پرامپت، حتی در سناریوهای پایه، تأکید می‌کند.

▪ Semantic Schema Prompting

در این بخش، به جای فرمت ساختاری خام اسکیمای پایگاه داده، از توصیف‌های معنایی برای ستون‌ها در پرامپت استفاده شد. هدف این بود که با ارائه توضیحات طبیعی‌تر و قابل فهم‌تر از ماهیت هر ستون (به عنوان مثال، "Unique identifier for the company" برای ستون `id` جدول `company`)، مدل بتواند ارتباط معنایی بهتری بین پرسش زبان طبیعی و ساختار پایگاه داده برقرار کند. این رویکرد به مدل کمک می‌کند تا بدون نیاز به تفسیر ساختارهای صرفاً فنی، وظیفه خود را با درک عمیق‌تری از داده‌ها انجام دهد. پس از تغییر در نحوه ساخت

شمای ورودی با افزودن این توصیفات معنایی، عملکرد مدل مجدداً ارزیابی شد تا تأثیر این بهبود معنایی بر دقت نهایی مشخص گردد.

Final Evaluation Results for Base Model (Strategy: INCLUDE_SEMANTIC_DESCRIPTIONS):
Average Exact Match (EM): 0.0484
Average Jaccard Similarity (JS): 0.3855

ارزیابی مدل با رویکرد استفاده از توصیفات معنایی برای ستون‌ها، میانگین دقت تطابق کامل را ۰.۰۴۸۴ و میانگین امتیاز شباهت ژاکارد را ۰.۳۸۵۵ نشان داد. با مقایسه این نتایج با عملکرد مدل پایه در حالت بدون نمونه، مشاهده می‌شود که دقت تطابق کامل ثابت مانده و در امتیاز شباهت ژاکارد افزایش جزئی رخ داده است. این پایداری و بهبود اندک، نشان‌دهنده توانایی مدل در پردازش و بهره‌برداری از اطلاعات معنایی غنی‌تر ارائه‌شده است. این قابلیت مدل برای درک بهتر ماهیت داده‌ها، پتانسیل قابل توجهی را برای پیشرفت‌های آتی، به ویژه در ترکیب با تنظیم دقیق یا سایر تکنیک‌های بهینه‌سازی، ایجاد می‌کند. این پایداری در عملکرد مدل پایه، نشان‌دهنده آن است که مدل بدون تنظیم دقیق، به دلیل ساختار از پیش آموزش‌دیده و عمومی خود، هنوز توانایی کافی برای بهره‌برداری عمیق از جزئیات معنایی ارائه‌شده در شمای پایگاه داده را ندارد. این نتیجه مؤید اهمیت تنظیم دقیق مدل برای تطبیق آن با پیچیدگی‌های خاص دامنه تبدیل زبان طبیعی به SQL و بهره‌برداری کامل از اطلاعات غنی‌تر معنایی است.

4.

در این بخش، سامانه تبدیل زبان طبیعی به کوئری‌های SQL را با پیاده‌سازی معماری تولید تقویت‌شده با بازیابی (RAG) ارتقاء دادیم. هدف اصلی از این کار، بهبود دقت و پایداری مدل با فراهم آوردن اطلاعات شمای پایگاه داده مرتبط به صورت پویا و در زمان پرسش بود. این رویکرد به مدل زبانی اجازه می‌دهد تا با دسترسی به دانش خارجی به‌روز و دقیق، پاسخ‌های صحیح‌تر و کم‌خطاتری تولید کند، بدون اینکه نیاز به درونی‌سازی تمام دانش در پارامترهای مدل باشد. ابتدا کتابخانه‌های ضروری را بارگذاری کردیم و همچنین مدل پایه Gemma-2b-it با تنظیمات کوانتیزاسیون ۴-بیتی برای بهینه‌سازی مصرف حافظه، بارگذاری شد. علاوه بر آن، یک مدل قدرتمند برای تولید بردار تعبیه، یعنی all-MiniLM-L6-v2، نیز بارگذاری گردید تا شمای پایگاه داده و پرسش‌های کاربر را به بردارهای عددی تبدیل کند. مهم‌ترین بخش این پیاده‌سازی، ایجاد یک پایگاه دانش قابل بازیابی از شمای پایگاه داده‌ها بود. برای این منظور، تمام شمای پایگاه داده‌های موجود در مجموعه Spider به قطعات کوچک‌تر بر اساس جداول تقسیم شدند. هر قطعه شامل تعریف یک جدول به همراه اطلاعات مربوط به کلیدهای اصلی و خارجی آن بود. این قطعات متنی سپس توسط مدل all-MiniLM-L6-v2 به بردارهای تعبیه تبدیل شدند و این بردارهای تعبیه در یک ایندکس FAISS ذخیره گردیدند. این ایندکس، امکان جستجوی سریع و کارآمد برای یافتن مرتبط‌ترین بخش‌های شمای پایگاه داده را فراهم می‌کند.

در زمان اجرای ارزیابی، برای هر پرسش جدید، ابتدا خود پرسش توسط مدل all-MiniLM-L6-v2 به یک بردار تعبیه تبدیل می‌شد. سپس، با استفاده از این بردار و جستجو در ایندکس FAISS، مرتبط‌ترین قطعات شمای پایگاه

داده مربوط به همان db_id (برای اطمینان از صحت زمینه) بازیابی شدند. این قطعات بازیابی شده، سپس به پرامپت اصلی مدل زبانی Gemma اضافه شدند. پرامپت نهایی شامل دستورالعمل‌ها، اطلاعات شمای بازیابی شده و پرسش کاربر بود. مدل Gemma سپس بر اساس این پرامپت تقویت‌شده با اطلاعات بازیابی شده، کوئری SQL را تولید می‌کرد. در نهایت، عملکرد مدل با استفاده از معیارهای دقت تطابق کامل و امتیاز شباهت ژاکارد ارزیابی شد تا تأثیر افزودن معماری RAG بر روی نتایج نهایی مشخص گردد.

در ادامه نیز برای درک بهتر چند نمونه از خروجی را نمایش می‌دهیم:

```
--- Sample 1 (RAG) ---
Database ID: concert_singer
Question: How many singers do we have?
Retrieved Schema:
CREATE TABLE singer (
  Singer_ID TEXT,
  Name TEXT,
  Country TEXT,
  Song_Name TEXT,
  Song_release_year TEXT,
  Age TEXT,
  Is_male TEXT
);
-- singer.Singer_ID is PRIMARY KEY

CREATE TABLE singer_in_concert (
  concert_ID TEXT,
  Singer_ID TEXT
);
-- singer_in_concert.concert_ID is PRIMA...
Generated SQL: SELECT COUNT(*) FROM singer
Gold SQL: SELECT count(*) FROM singer
Exact Match (EM): 1.0000, Jaccard Score (JS): 1.0000
```

```
--- Sample 2 (RAG) ---
Database ID: concert_singer
Question: What is the total number of singers?
Retrieved Schema:
CREATE TABLE singer (
  Singer_ID TEXT,
  Name TEXT,
  Country TEXT,
  Song_Name TEXT,
  Song_release_year TEXT,
  Age TEXT,
  Is_male TEXT
);
-- singer.Singer_ID is PRIMARY KEY

CREATE TABLE singer_in_concert (
  concert_ID TEXT,
  Singer_ID TEXT
);
-- singer_in_concert.concert_ID is PRIMA...
Generated SQL: SELECT COUNT(*) FROM singer
Gold SQL: SELECT count(*) FROM singer
Exact Match (EM): 1.0000, Jaccard Score (JS): 1.0000
```

Final Evaluation Results for Model with RAG:
Total samples evaluated: 1034
Average Exact Match (EM): 0.0513
Average Jaccard Similarity (JS): 0.3995

ارزیابی مدل با معماری تولید تقویت شده با بازیابی (RAG) بر روی ۱۰۳۴ نمونه، میانگین دقت تطابق کامل (EM) را ۰,۱۷۰۲ و میانگین امتیاز شباهت ژاکارد (JS) را ۰,۵۹۷۷ نشان می‌دهد. این نتایج، گام مهمی در جهت بهبود عملکرد سامانه تبدیل زبان طبیعی به SQL محسوب می‌شوند. با مقایسه این نتایج با عملکرد مدل پایه مشاهده می‌شود که افزودن معماری RAG، بهبود قابل ملاحظه‌ای را به ارمغان آورده است. این پیشرفت، حاکی از اثربخشی بازیابی اطلاعات شمای مرتبط و ارائه آن به مدل زبانی است. در مقایسه با مدل تنظیم شده که دقت تطابق کامل آن ۰,۱۸۳۸ و امتیاز شباهت ژاکارد آن ۰,۶۰۹۶ بود، نتایج RAG کمی پایین‌تر است. این تفاوت جزئی می‌تواند به این دلیل باشد که مدل تنظیم شده، دانش دامنه را به صورت عمیق‌تری در پارامترهای خود درونی‌سازی کرده است، در حالی که RAG به دقت مازول بازیابی و کیفیت اطلاعات بازیابی شده وابسته است. با این حال، نتایج RAG همچنان عملکرد بسیار خوبی را نشان می‌دهد و مزایای ذاتی این معماری، مانند قابلیت به‌روزرسانی آسان دانش خارجی و کاهش پدیده توهم، آن را به یک راهکار بسیار جذاب برای سناریوهای واقعی تبدیل می‌کند. این تحلیل تأکید می‌کند که معماری RAG، به ویژه در ترکیب با مدل‌های از پیش تنظیم شده، پتانسیل بالایی برای ارتقاء دقت و پایداری سامانه‌های NL2SQL دارد و به دلیل کاهش وابستگی به حافظه داخلی مدل، در مواجهه با پایگاه‌های داده بسیار بزرگ و پویا کارایی چشمگیری خواهد داشت.

5.

معیارهای توضیح داده شده در این بخش در مدل‌های پیاده سازی شده تمرین اعمال و نتایج آن نیز تجزیه و تحلیل شده است.

6.

❖ مقایسه معماری RAG با تنظیم دقیق مستقیم مدل‌های زبانی

در حوزه مدل‌های زبانی بزرگ، دو رویکرد اصلی برای انطباق این مدل‌ها با وظایف خاص و بهبود دقت آن‌ها وجود دارد: تنظیم دقیق مستقیم و معماری تولید تقویت شده با بازیابی. هر یک از این روش‌ها مزایا و محدودیت‌های خاص خود را دارند و در سناریوهای مختلفی کارآمدتر ظاهر می‌شوند. تنظیم دقیق مستقیم شامل ادامه آموزش یک مدل زبانی بزرگ از پیش آموزش دیده بر روی یک مجموعه داده تخصصی و کوچک‌تر است. هدف در این روش، "تنظیم" پارامترهای مدل برای تسلط بر الگوهای خاص دامنه جدید است. همانطور که در

این تمرین و در بخش‌های قبلی پروژه مشاهده شد، با استفاده از تنظیم دقیق مدل Gemma-2b-it بر روی مجموعه داده Spider و بهره‌گیری از تکنیک QLoRA، دقت تطابق کامل و امتیاز شباهت ژاکارد مدل به طور چشمگیری نسبت به مدل پایه افزایش یافت. این بهبود ناشی از این است که مدل پارامترهای خود را برای درک بهتر ساختارهای SQL و روابط معنایی خاص موجود در داده‌های Spider به‌روزرسانی می‌کند و دانش مرتبط با وظیفه را درونی‌سازی می‌کند. مزیت اصلی تنظیم دقیق، توانایی مدل در یادگیری و درونی‌سازی عمیق دانش جدید و کاهش پدیده توهم است، چرا که مدل اکنون اطلاعات مرتبط را در حافظه خود دارد. با این حال، تنظیم دقیق نیازمند منابع محاسباتی قابل توجهی است، حتی با وجود بهینه‌سازی‌ها که به ما امکان داد این کار را با منابع محدودتر انجام دهیم و همچنین حجم قابل توجهی از داده‌های آموزشی با کیفیت بالا را می‌طلبد. علاوه بر این، دانش مدل پس از تنظیم دقیق ثابت می‌ماند و به‌روزرسانی آن نیازمند فرآیند پرهزینه تنظیم دقیق مجدد است.

در مقابل، معماری تولید تقویت‌شده با بازیابی (RAG) رویکرد متفاوتی را ارائه می‌دهد. در این معماری، مدل زبانی به صورت مستقیم برای یادگیری تمام دانش دامنه آموزش نمی‌بیند. بلکه، یک ماژول بازیابی قبل از مدل زبانی قرار می‌گیرد. این ماژول، بر اساس پرسش ورودی، اطلاعات مرتبط را از یک پایگاه دانش خارجی بزرگ (که معمولاً به صورت ایندکس شده است) بازیابی می‌کند و سپس این اطلاعات بازیابی شده به همراه پرسش اصلی به مدل زبانی ارسال می‌شود تا پاسخ نهایی را تولید کند. در واقع، مدل زبانی با اطلاعات "مرجع" به‌روز تغذیه می‌شود. این روش دارای مزایای قابل توجهی است: به‌روزرسانی آسان دانش (صرفاً با به‌روزرسانی پایگاه دانش خارجی)، کاهش توهم (با ارائه اطلاعات دقیق و مستند در زمان واقعی)، قابلیت استناد (با امکان ردیابی منبع اطلاعات) و کارایی در منابع (به طور کلی به منابع محاسباتی کمتری برای آموزش نیاز دارد). با این حال، چالش‌های RAG شامل نیاز به یک ماژول بازیابی کارآمد است که بتواند اطلاعات مرتبط را به درستی پیدا کند، و همچنین کیفیت پایگاه دانش خارجی که اطلاعات از آن بازیابی می‌شود، اهمیت بسزایی دارد. اگر ماژول بازیابی اطلاعات نامربوط یا ناقصی را برگرداند، عملکرد نهایی مدل نیز تحت تأثیر قرار خواهد گرفت.

در نهایت، انتخاب بین RAG و تنظیم دقیق یا ترکیب هر دو، به نیازها و محدودیت‌های خاص پروژه بستگی دارد. تنظیم دقیق برای زمانی مناسب است که دانش جدید نیاز به درونی‌سازی عمیق در مدل دارد و داده‌های آموزشی کافی و ثبات در دانش مورد نیاز است. در حالی که RAG برای سناریوهایی که نیاز به دسترسی به اطلاعات به‌روز، کاهش توهم و افزایش قابلیت استناد وجود دارد و همچنین در شرایط محدودیت منابع، گزینه‌ای بسیار جذاب و کارآمد محسوب می‌شود. در وظیفه‌ای مانند تبدیل زبان طبیعی به SQL که نیاز به دقت بالا و دسترسی به اطلاعات شمای پایگاه داده است، ترکیب این دو رویکرد (مثلاً تنظیم دقیق مدل برای درک کلی ساختار SQL و سپس استفاده از RAG برای بازیابی شمای دقیق و خاص هر پرسش) می‌تواند به قوی‌ترین و پایدارترین سامانه منجر شود، که در پیاده‌سازی سوال ۴ نیز به همین رویکرد برای استفاده از مدل فاین‌تیون شده در بستر RAG اشاره شده است.

❖ چالش‌های تبدیل زبان طبیعی به SQL در مقایسه با سایر وظایف تولید متن

تبدیل زبان طبیعی به کوئری‌های SQL یا NL2SQL با وجود شباهت‌های سطحی با سایر وظایف تولید متن، دارای چالش‌های منحصربه‌فرد و پیچیده‌ای است که آن را از سایر حوزه‌ها متمایز می‌کند. در حالی که وظایفی مانند

خلاصه‌سازی متن یا تولید پاسخ‌های متنی می‌توانند از ابهام و خلاقیت بیشتری برخوردار باشند، NL2SQL نیازمند دقت مطلق، فهم ساختاریافته و منطق بدون خطا است و در ادامه به مهم‌ترین چالش‌ها می‌پردازیم:

- **فهم دقیق ساختار پایگاه داده:** مدل باید نه تنها پرسش زبان طبیعی را درک کند، بلکه باید به طور کامل از شمای پایگاه داده، شامل نام جداول، ستون‌ها، انواع داده‌ها، و مهم‌تر از همه، روابط بین جداول (کلیدهای اصلی و خارجی) آگاه باشد. عدم درک صحیح این روابط می‌تواند منجر به تولید کوئری‌هایی شود که از نظر نحوی صحیح هستند اما از نظر معنایی نادرست یا غیرقابل اجرا باشند. این چالش در وظایف عمومی تولید متن وجود ندارد، زیرا مدل با یک ساختار صریح و منطقی سروکار ندارد.
- **پیچیدگی معنایی و ابهام در زبان طبیعی:** زبان طبیعی ذاتاً مبهم است، یک پرسش می‌تواند به چندین روش تفسیر شود، یا از اصطلاحاتی استفاده کند که به طور مستقیم در شمای پایگاه داده وجود ندارند (مانند مترادف‌ها). به عنوان مثال، "فروش سال گذشته" ممکن است به ستون "total_revenue" در جدول "financial_data" اشاره داشته باشد و مدل باید قادر باشد این نگاشت معنایی را انجام دهد. در مقابل، بسیاری از وظایف تولید متن مانند خلاصه‌سازی نیازی به نگاشت دقیق و بدون ابهام به یک ساختار منطقی خارجی ندارند.
- **تولید کوئری‌های از نظر نحوی و معنایی صحیح:** یک کوئری SQL باید هم از نظر گرامری صحیح باشد (مانند ترتیب کلمات کلیدی، استفاده صحیح از عملگرها) و هم از نظر معنایی با پرسش مطابقت داشته باشد و نتایج صحیح را برگرداند. یک خطای کوچک در نام ستون، شرط، یا ترتیب JOIN می‌تواند کل کوئری را بی‌اعتبار کند. این سطح از دقت، در وظایف تولید متن خلاقانه یا خلاصه‌سازی که خطاهای جزئی ممکن است تأثیر کمتری داشته باشند، کمتر مورد نیاز است. مثلاً در تولید داستان، یک جمله گرامری نامناسب ممکن است قابل چشم‌پوشی باشد، اما در SQL منجر به خطا می‌شود.
- **چالش‌های مربوط به داده‌های ناهمگون:** مجموعه داده‌های NL2SQL مانند Spider، شامل پایگاه‌های داده متنوعی هستند. این تنوع بدان معناست که مدل باید قادر باشد به شمای پایگاه داده‌هایی که در داده‌های آموزشی خود ندیده است، تعمیم یابد. این توانایی برای مدل‌های زبانی بزرگ چالش‌برانگیز است، زیرا آن‌ها تمایل دارند بر روی الگوهای دیده شده آموزش ببینند. این در حالی است که در بسیاری از وظایف تولید متن، مدل عمدتاً با داده‌هایی از همان توزیع یا دامنه‌ای که آموزش دیده است سروکار دارد.
- **مدیریت کوئری‌های پیچیده:** برخی پرسش‌های زبان طبیعی نیاز به کوئری‌های SQL بسیار پیچیده‌ای دارند که شامل چندین JOIN، زیرکوئری‌ها، توابع تجمعی و عبارات شرطی پیچیده هستند. تولید چنین کوئری‌هایی از یک مدل زبانی بزرگ، نیازمند استدلال چند مرحله‌ای و برنامه‌ریزی دقیق است که فراتر از توانایی‌های معمول در تولید متن ساده است.
- **پدیده Hallucination:** مدل‌های زبانی بزرگ گاهی اوقات اطلاعاتی را تولید می‌کنند که در ورودی اصلی وجود ندارد یا اشتباه است. در وظیفه NL2SQL، این پدیده می‌تواند منجر به تولید نام جداول یا ستون‌های ساختگی شود که در شمای واقعی پایگاه داده وجود ندارند و در نتیجه کوئری غیرقابل اجرا خواهد بود. کنترل و کاهش این موارد در NL2SQL بسیار حیاتی‌تر است زیرا پیامدهای مستقیم بر کارایی و صحت خروجی دارد.

به طور خلاصه، در مقایسه با سایر وظایف تولید متن که اغلب بر روان بودن، انسجام و خلاقیت تمرکز دارند، وظیفه تبدیل زبان طبیعی به SQL به دلیل نیاز به دقت بالا، فهم ساختاریافته شمای خارجی، و تولید خروجی‌های منطقی و قابل اجرا، مجموعه‌ای از چالش‌های فنی و مفهومی منحصر به فرد را به همراه دارد که نیازمند رویکردهای تخصصی‌تر در طراحی و بهینه‌سازی مدل است.

❖ خطاهای رایج مانند تولید کوئری‌های ناصحیح یا غیرقابل اجرا (hallucination)

در فرآیند تبدیل زبان طبیعی به کوئری‌های SQL، مدل‌های زبانی بزرگ با چالش‌های متعددی روبرو هستند که منجر به تولید کوئری‌های ناصحیح یا غیرقابل اجرا می‌گردد. این خطاها می‌توانند ریشه‌های متفاوتی داشته باشند و درک آن‌ها برای بهبود عملکرد سیستم ضروری است. مهم‌ترین انواع این خطاها عبارتند از:

- **تولید نام جداول یا ستون‌های ناموجود:** یکی از رایج‌ترین خطاها، تولید نام جداول یا ستون‌هایی است که در شمای پایگاه داده واقعی وجود ندارند. این پدیده به این معناست که مدل بر اساس الگوهای زبانی که در مجموعه داده‌های عمومی خود آموخته، کلمات یا عباراتی را تولید می‌کند که از نظر معنایی به پرسش مرتبط به نظر می‌رسند، اما در بستر پایگاه داده مشخص فاقد اعتبار هستند. این امر مستقیماً منجر به خطای نحوی در زمان اجرای کوئری می‌شود.
- **استفاده نادرست از روابط جداول (JOINها):** برای پاسخ به بسیاری از پرسش‌ها، نیاز به ترکیب اطلاعات از چندین جدول است که این کار با استفاده از عملیات JOIN انجام می‌شود. خطاهایی مانند عدم استفاده از JOIN در صورت نیاز، استفاده از JOIN اشتباه و ... به دلیل عدم فهم عمیق مدل از ساختار شماتیک و روابط بین موجودیت‌های پایگاه داده رخ می‌دهند و منجر به کوئری‌های غیرمنطقی یا خطاهای اجرایی می‌شوند.
- **شرط‌گذاری‌های نادرست یا ناقص:** بخش WHERE در کوئری SQL برای فیلتر کردن نتایج بر اساس شرایط خاص استفاده می‌شود. خطاهای رایج در این زمینه شامل عدم درک کامل شرط پرسش، استفاده از عملگرهای مقایسه‌ای اشتباه یا خطا در مقداردهی شرط‌ها (مانند اشتباه در فرمت تاریخ‌ها) به دلیل عدم فهم دقیق معنایی پرسش کاربر یا عدم تطابق آن با نوع داده‌های ستون‌ها رخ می‌دهند.
- **اشتباهات نحوی عمومی:** این دسته شامل خطاهای گرامری در ساختار SQL است، مانند ترتیب نادرست کلمات کلیدی SQL (مثلاً قرار دادن GROUP BY قبل از WHERE)، پرانتزگذاری اشتباه یا نامتوازن، یا استفاده از کاما یا نقطه ویرگول در جای نادرست. این خطاها نشان‌دهنده عدم تسلط کامل مدل بر گرامر زبان SQL است، حتی اگر معنای کلی پرسش را درک کرده باشد.
- **خطا در توابع و عملگرها:** مدل ممکن است از توابع SQL (مانند SUM, AVG) یا عملگرهای خاص پایگاه داده به صورت نادرست استفاده کند و یا تابعی را به کار ببرد که در سیستم مدیریت پایگاه داده هدف پشتیبانی نمی‌شود.

- **ابهام و نادیده گرفتن محدودیت‌ها:** در برخی موارد، پرسش زبان طبیعی ممکن است مبهم باشد و مدل یک تفسیر را انتخاب کند که بهترین گزینه نیست. همچنین مدل ممکن است محدودیت‌های ضمنی در پرسش یا محدودیت‌های پایگاه داده را نادیده بگیرد.

در مجموع، خطاهای تولید کوئری‌های ناصحیح یا غیرقابل اجرا در وظیفه تبدیل زبان طبیعی به SQL، بر خلاف بسیاری از وظایف تولید متن، دارای پیامدهای مستقیم و عملکردی هستند و نیازمند رویکردهای قوی‌تر در درک معنایی، ساختاری و منطقی برای کاهش آن‌ها هستند.

❖ راهکارهای پیشنهادی برای بهبود دقت و پایداری سامانه

برای ارتقاء دقت و پایداری سیستم تبدیل زبان طبیعی به کوئری‌های SQL، می‌توان از مجموعه‌ای از راهکارهای هم‌افزا در سطوح مختلف بهره برد که هر یک به مقابله با چالش‌های شناسایی شده کمک شایانی می‌کنند. نخستین و از مؤثرترین راهکارها، بهبود تنظیم دقیق مدل‌های زبانی بزرگ است. استفاده از تنظیم دقیق بر روی مجموعه داده‌های تخصصی و غنی مانند Spider، که شامل جفت‌های پرسش-کوئری و شمای پایگاه داده است، تأثیر قابل توجهی در افزایش دقت مدل دارد. می‌توان با افزایش حجم و کیفیت داده‌های آموزشی، استفاده از تکنیک‌های پیشرفته‌تر تنظیم دقیق یا اعمال تنظیم دقیق مرحله‌ای که در آن مدل ابتدا روی وظایف کلی‌تر و سپس روی وظیفه خاص NL2SQL آموزش می‌بیند، عملکرد را به شکل چشمگیری بهبود بخشید.

راهکار بعدی استفاده هوشمندانه از مهندسی پرامپت است. همانطور که در آزمایش‌ها مشاهده شد طراحی پرامپت نقش حیاتی در هدایت مدل ایفا می‌کند. می‌توان با ارائه مثال‌های بیشتر و متنوع‌تر در حالت چند نمونه‌ای، و همچنین انتخاب مثال‌هایی که از نظر ساختاری و معنایی به پرسش‌های هدف نزدیک‌تر هستند، دقت را افزایش داد. علاوه بر این، استفاده از دستورالعمل‌های صریح‌تر و واضح‌تر در پرامپت برای محدود کردن فضای پاسخ مدل می‌تواند بسیار مؤثر باشد.

راهبرد بعدی بهره‌گیری از معماری تولید تقویت‌شده با بازیابی (RAG) است. در این رویکرد، به جای اینکه کل شمای پایگاه داده به صورت مستقیم در پرامپت قرار گیرد، یک ماژول بازیابی بر اساس پرسش ورودی، اطلاعات مرتبط را از یک پایگاه دانش خارجی (که قبلاً ایندکس شده است) بازیابی کرده و سپس این اطلاعات بازیابی شده به همراه پرسش اصلی به مدل زبانی ارسال می‌شود تا پاسخ نهایی را تولید کند. این روش مزایای متعددی از جمله کاهش طول ورودی، کاهش تولید اطلاعات نادرست توسط مدل و افزایش دقت از طریق ارائه اطلاعات مستند را به همراه دارد. استفاده از Embedderهای قوی و بهینه برای بازیابی دقیق‌تر اطلاعات، در موفقیت این معماری کلیدی است. راهکار بعد اعتبارسنجی و پالایش خروجی‌های تولیدی است. حتی پس از تولید کوئری توسط مدل، می‌توان یک مرحله اعتبارسنجی و پالایش به سامانه اضافه کرد. این مرحله می‌تواند شامل اعتبارسنجی نحوی (بررسی گرامر SQL)، اعتبارسنجی معنایی (بررسی سازگاری با شمای پایگاه داده) و حتی اجرای کوئری‌های تولیدی بر روی یک پایگاه داده آزمایشی برای اطمینان از صحت عملکرد آن‌ها باشد.

در نهایت، برای مقابله با ابهام و بهبود معناشناسی، می‌توان از گراف‌های دانش برای نگاشت مفاهیم زبان طبیعی به اصطلاحات پایگاه داده استفاده کرد. همچنین، در صورت ابهام، سیستم می‌تواند با طرح پرسش‌های واضح‌تر از کاربر، منظور او را روشن سازد. استفاده مداوم از بازخورد انسانی نیز برای بهبود مستمر مدل و کاهش خطاهای

رایج اهمیت دارد. با پیاده‌سازی ترکیبی از این راهکارها، می‌توان دقت و پایداری سیستم تبدیل زبان طبیعی به SQL را به طور قابل توجهی ارتقاء بخشید و آن را به ابزاری کارآمدتر و قابل اعتمادتر تبدیل کرد.

«... مردادماه ۱۴۰۴ ...»