Text to sql

Arabic Spyder FineTuning using motherduckdb/DuckDB-NSQL-7B-v0.1

Names :

Rahma Mohamed 320210244

Mohamed Said 320210255

Rahma: https://huggingface.co/Rahmaa33/MotherDuckTEXT2SQLLArabicc

Introduction:

The Arabic text-to-SQL problem involves developing a system that can interpret the linguistic nuances of Arabic natural-language queries, then generate accurate SQL statements aligned with the target database schema, while taking into account Arabic morphology, orthographic variations, and the scarcity of dedicated resources in this field. In our approach, we utilized the Spider dataset, along with its corresponding database schemas, to provide a foundation for training and evaluating Arabic text-to-SQL models. We experimented with various sequence-to-sequence models, and ultimately achieved the best results using the motherduckdb/DuckDB-NSQL-7B-v0.1 model. After fine-tuning, this model reached a Final Execution Accuracy of 69.65% and a Final BLEU Score of 0.4698 on our Arabic-translated dataset.

Literature Review

Ar-Spider: Text-to-SQL in Arabic

Authors: Saleh Almohaimeed, Saad Almohaimeed, Mansour Al Ghanim, Liqiang Wang

Published: February 22, 2024

Summary: This paper introduces Ar-Spider, the first Arabic cross-domain text-to-SQL dataset. It addresses challenges unique to the Arabic language, such as schema linguistic and SQL structural issues. The authors evaluate two baseline models, LGESQL and S2SQL, using cross-lingual approaches to mitigate these challenges. The study reports decent performance, with LGESQL achieving 65.57% accuracy and S2SQL 62.48%, highlighting a performance gap compared to English datasets.

Read the full paper

AraSpider: Democratizing Arabic-to-SQL

Authors: Ahmed Heakl, Youssef Mohamed, Ahmed B. Zaky

Published: February 12, 2024

Summary: This study presents AraSpider, an Arabic version of the Spider dataset, aiming to enhance NLP capabilities for the Arabic-speaking community. The authors evaluate four multilingual translation models for English-to-Arabic translation and assess two models for generating SQL queries from Arabic text. The findings emphasize the effectiveness of back translation strategies and the

incorporation of contextual schema to improve model performance in Arabic NLP tasks.
Read the full paper

Dataset:

Spider is a large-scale, complex, and cross-domain dataset designed to train and evaluate models that convert natural language questions into SQL queries. It was introduced in 2018 by researchers from Yale University.

- Over 10,000 questions
- Covering 200 different databases
- Over 6000 translated Questions to Arabic
- Across 138 different domains (e.g., academic, sports, business, etc.)
- Each database contains multiple related tables, making queries more complex (joins, nested queries, aggregations, etc.)

DuckDB-NSQL-7B: Specialized SQL Generation Model

1. Model Overview

DuckDB-NSQL-7B is part of the NSQL family,7 billion parameters and 32 transformer layers, a set of autoregressive open-source foundation models designed for SQL generation. It is based on Meta's Llama-2 7B model, which has been further pre-trained on general SQL queries and fine-tuned specifically on DuckDB text-to-SQL pairs. Unlike many text-to-SQL models, it can generate **any valid DuckDB SQL statement**, not just SELECT queries.

2. Training Data

The model was trained on **200,000 DuckDB text-to-SQL pairs**, which were synthetically generated using **Mixtral-8x7B-Instruct-v0.1** and guided by the **DuckDB v0.9.2 documentation**. Additionally, it incorporates text-to-

SQL pairs from **NSText2SQL**, which were transpiled to DuckDB SQL using **sqlglot**, ensuring compatibility with the DuckDB engine.

3. Evaluation

To measure its performance, the model was tested on a **DuckDB-specific benchmark** consisting of **75 text-to-SQL pairs**. This benchmark helps assess the accuracy and reliability of SQL query generation tailored to DuckDB's syntax and functionalities. The dataset is publicly available for further evaluation and improvements.

4. Training Procedure

The model was trained using **cross-entropy loss**, optimizing the likelihood of generating correct SQL sequences. During fine-tuning, the loss was computed **only on the SQL portion** of the input-output pairs. Training was performed on **80GB A100 GPUs**, utilizing **data and model parallelism** for efficient processing. The model was fine-tuned for **500 epochs** to achieve optimal performance.

5. Intended Use & Limitations

DuckDB-NSQL-7B is designed for **text-to-SQL generation**, making it useful for applications where users need to query databases using natural language. It performs best when given **table schemas and well-structured prompts**. Unlike some text-to-SQL models, it supports a **wide range of DuckDB SQL statements**, including those related to **official DuckDB extensions**. However, its accuracy depends on how closely the input follows the expected prompt format.

The main Model

motherduckdb/DuckDB-NSQL-7B-v0.1

```
model_name = "motherduckdb/DuckDB-NSQL-7B-v0.1"  # Specialized for SQL generation

# Configure 4-bit quantization for memory efficiency
bnb_config = BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_use_double_quant=True,
    bnb_4bit_quant_type="nf4",
    bnb_4bit_compute_dtype=torch.float16
)
```

The Hyperparameter

```
[ ]  from transformers import TrainingArguments, Trainer, DataCollatorForLanguageModeling

    # 1. Configure training with padding control
    training_args = TrainingArguments(
        output_dir="./arabic_sql_results",
        num_train_epochs=1,                    # Reduce from 3 → 1 epoch (you can increase later)
        per_device_train_batch_size=4,         # Increase from 2 → 4 (if GPU memory allows)
        gradient_accumulation_steps=2,         # Reduce from 4 → 2 (faster updates)
        learning_rate=3e-5,                    # Slightly higher LR for faster convergence
        fp16=True,                             # Keep mixed precision
        save_steps=500,                        # Save less frequently
        logging_steps=25,                      # Log more frequently for monitoring
        optim="adamw_torch_fused",             # Faster AdamW implementation
        report_to="none",
        max_steps=500,                         # Limit to 500 steps for quick testing
        warmup_ratio=0.1,                      # Helps stabilize training
        lr_scheduler_type="cosine",            # Better learning rate decay
    )
```

The prompt

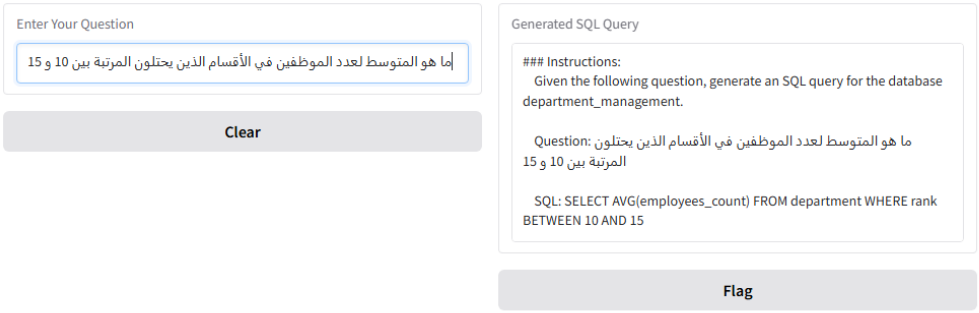
```
def create_arabic_sql_prompt(item):
    """Clean prompt format without extra padding"""
    return f"""### السؤال:
{item['arabic']}

### قاعدة البيانات:
{item['db_id']}

### SQL:
{item['query']}"""

# 1. Create clean datasets
train_data = [{"text": create_arabic_sql_prompt(item)} for item in spider_data[:5000]]
val_data = [{"text": create_arabic_sql_prompt(item)} for item in spider_data[5000:]]
```

The Evaluation Part

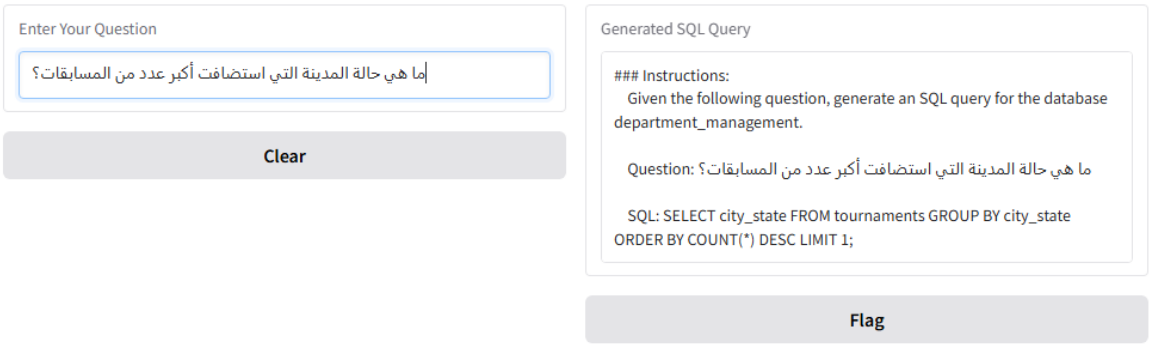
```
Final Execution Accuracy: 69.65%
Final BLEU Score: 0.4698313105174533
```

The Test Part

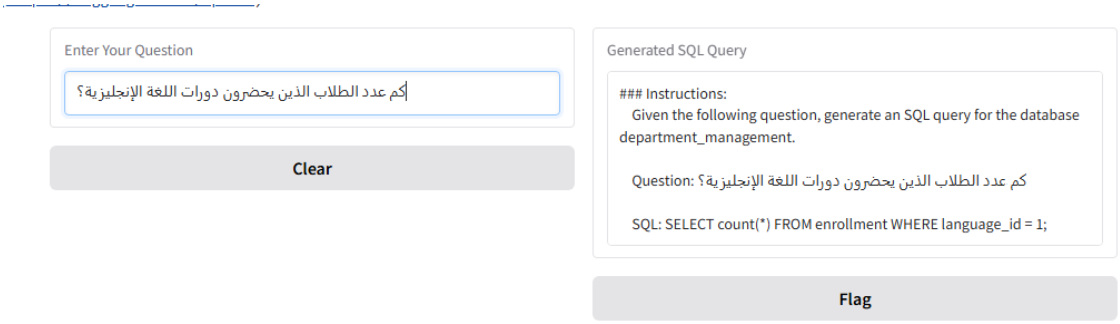
The Ex1:



The Ex2:



The Ex3:



The Ex4 :