Fine-Tuning Microsoft's Phi-2 Model for SQL Query Generation

Nemanja Mitrić

Fakulteta za elektrotehniko, računalništvo in informatiko, Maribor

Abstract

This report presents the process and results of fine-tuning Microsoft's Phi-2 small language model for generating SQL queries using the Spider dataset. The methodology leverages QLoRA and PEFT for efficient training, achieving improved performance over the baseline model. Key findings include quantitative improvements in ROUGE metrics and qualitative enhancements in query accuracy.

Contents

1 Introduction

Small language models (SLMs) like Microsoft's Phi-2, with 2.7 billion parameters, offer a resource-efficient alternative to Large language models which are resource intensive to train and run. This project focuses on fine-tuning Phi-2 for SQL query generation. Using the Spider dataset, we aim to enhance the model's ability to generate accurate SQL queries from natural language questions leveraging efficient fine-tuning techniques like QLoRA and PEFT to achieve this on modest hardware.

2 Setup

2.1 Problem Definition

The task is to fine-tune Phi-2 to generate SQL queries from natural language questions. This involves adapting the model to understand database schemas and produce syntactically correct SQL. The Spider dataset, containing natural language questions paired with SQL queries across various databases, serves as the training foundation.

2.2 Dataset Description

The Spider dataset from Hugging Face "xlangai/spider" is used for training and evaluation. It is made up of 7,000 training examples and 1,034 validation examples, each including a database ID, natural language question, and the SQL query. No extensive preprocessing was required, as the dataset is well-structured, though minor formatting ensured compatibility with the tokenizer.

db_id string · *classes*	query string · *lengths*	question string · *lengths*
concert_si… 4.4%	20…61 23%	50…66 28.2%
concert_singer	SELECT count(*) FROM singer	How many singers do we have?
concert_singer	SELECT count(*) FROM singer	What is the total number of singers?
concert_singer	SELECT name , country , age FROM singer ORDER BY age DESC	Show name, country, age for all singers ordered by age…
concert_singer	SELECT name , country , age FROM singer ORDER BY age DESC	What are the names, countries, and ages for every singer in…

3 Experiment

3.1 Methodology

The fine-tuning process employs QLoRA (Quantized Low-Rank Adaptation) and PEFT (Parameter-Efficient Fine-Tuning) to optimize Phi-2's performance on a single GPU. The model is quantized to 4 bits using BitsAndBytesConfig, reducing memory usage. Training involves adapting the pre-trained Phi-2 model to the Spider dataset, focusing on generating accurate SQL queries from natural language inputs. This approach minimizes computational overhead while maximizing task-specific performance.

3.2 Experiment Setup

The experiments were conducted in a Kaggle notebook environment with GPU acceleration to support the computational demands of fine-tuning. The setup process included the following components:

- **Environment Configuration**: Key libraries were installed to facilitate model training and evaluation. This includes "bitsandbytes" for quantization, "transformers==4.47.0" for model handling, "peft" for parameter-efficient fine-tuning, "accelerate" for optimized training, "datasets" for data loading, "scipy" and evaluate for metrics computation, "trl" for training utilities, and "rouge_score" for evaluation.
- **Model Loading**: The pre-trained Phi-2 model (`microsoft/phi-2`) was loaded with 4-bit quantization enabled via BitsAndBytesConfig. Initial GPU memory usage was approximately 10024 MB.
- **Tokenizer Setup**: The `AutoTokenizer` was initialized with settings to optimize memory usage during training. A separate evaluation tokenizer was configured with use_fast=False. The pad token was set to the EOS token for both.
- **Training Configuration**: The PEFT/LoRA model was set up with QLoRA, adding low-rank adapters to the model.

```
TrainingArguments(
    warmup_steps=1,
    per_device_train_batch_size=1,
    gradient_accumulation_steps=4,
    max_steps=1000,
    learning_rate=2e-4,
    optim="paged_adamw_8bit",
    logging_steps=25,
    save_strategy="steps",
    save_steps=25,
```

```
    eval_strategy="steps",
    eval_steps=25,
    do_eval=True,
    gradient_checkpointing=True,
    ...
)
```

- **Evaluation Framework**: Performance was evaluated using ROUGE metrics (ROUGE-1, ROUGE-2, ROUGE-L, ROUGE-Lsum) via the "evaluate" library, complemented by qualitative human evaluation of generated queries. A random seed of 42 was set to ensure reproducibility.

The setup enabled zero-shot inferencing to establish a baseline, followed by fine-tuning and post-training evaluation, all within the resource constraints of a single GPU.

4 Results

4.1 Gotten Results

The results are summarized as follows:

- **Quantitative Results**: The original model's ROUGE scores were:

  `ROUGE-1: 0.2290 ROUGE-2: 0.0284`
  `ROUGE-L: 0.1977 ROUGE-Lsum: 0.1925`

  After fine-tuning with PEFT/LoRA, the scores improved to:

  `ROUGE-1: 0.2985 ROUGE-2: 0.2339`
  `ROUGE-L: 0.2887 ROUGE-Lsum: 0.2910`

  This corresponds to absolute percentage improvements of:

  `6.95% (ROUGE-1)`
  `20.54% (ROUGE-2)`
  `9.10% (ROUGE-L)`
  `9.85% (ROUGE-Lsum).`

  These metrics, computed using the "rouge" evaluator indicate enhanced similarity between generated and reference queries.

- **Qualitative Results**: The fine-tuned model produced more accurate and contextually relevant SQL queries. For instance, for the query "List the number of singers per country," the baseline model produced an ASCII list of cities and singers, while the fine-tuned model generated a correct SQL Query `SELECT country, count(*) FROM singer GROUP BY country`

- **Error Analysis**: While the fine-tuned model improved overall, some generated results were still just sentences instead of SQL Queries. This is most likely due to the original model being focused on conversations more than anything.

4.2 Analysis

- **Empirical Analysis**: The significant improvement in ROUGE-2 (20.54%) suggests that the fine-tuned model better captures bigram overlaps, indicating improved syntactic accuracy and phrase alignment with reference queries. The ROUGE-L and ROUGE-Lsum gains (9.10% and 9.85%) reflect enhanced sequence similarity which is very important for SQL queries where token order is critical. The relatively modest ROUGE-1 improvement (6.95%) suggests that while unigram overlap improved, some queries still aren't perfect.

- **Descriptive Analysis**: Qualitatively, the fine-tuned model excelled in generating queries for straightforward aggregations (`COUNT`, `GROUP BY`) and filtering conditions (e`WHERE age > 56`).

- **Limitations**: The Spider dataset, while diverse, may introduce biases toward certain query types (e.g., aggregations over joins). Additionally, the model's reliance on 4-bit quantization, while memory-efficient, may have limited its capacity to capture nuanced patterns.
- **Future Improvements**: Further fine-tuning with augmented datasets or schema-specific preprocessing could enhance performance. A bigger dataset would also improve the learning.

5 Conclusions

This project successfully fine-tuned Microsoft's Phi-2 model for SQL query generation using the Spider dataset, leveraging QLoRA and PEFT for efficient training on a single GPU. The fine-tuned model achieved significant ROUGE score improvements and qualitative enhancements in query generating tasks. These findings underscore the potential of SLMs like Phi-2 for specialized NLP tasks, offering a resource-efficient alternative to LLMs. Future work could explore dataset augmentation, or algorithm optimization to further enhance performance.

References

References

[1] XLang NLP Lab, *Spider*, https://huggingface.co/datasets/xlangai/spider.

[2] Microsoft, *Phi-2 SLM*, https://huggingface.co/microsoft/phi-2

[3] Suman Das, *FineTune Phi-2 on Custom DataSet*, https://www.kaggle.com/code/dassum/finetune-phi-2-on-custom-dataset