Fine-Tuning LLM Using LoRA for Translating Natural Language Queries into Elasticsearch DSL

Abhinandnan Sakalakavanar
Dept. of CSE
PES University
Bengaluru, India

Adarsh Satish Upase
Dept. of CSE
PES University
Bengaluru, India

Achyuth K
Dept. of CSE
PES University
Bengaluru, India

Dr. Bhargavi
Faculty Guide, Dept. of CSE
PES University
Bengaluru, India

***Abstract*—Natural language querying enables users to interact with complex data systems using intuitive language rather than structured query syntax. While substantial research has been conducted on translating natural language into SQL for relational databases, there is limited work focused on document-oriented systems like Elasticsearch, which require queries written in a deeply nested Domain-Specific Language (DSL).In this work, we propose a parameter-efficient fine-tuning pipeline based on Low-Rank Adaptation (LoRA) to adapt large language models for the task of natural language to Elasticsearch DSL translation. LoRA enables us to fine-tune only a small subset of model parameters, significantly reducing memory and compute requirements while maintaining model performance. As no public dataset exists for this task, we construct a custom dataset of 4,000 examples by adapting and extending SQL query benchmarks with Elasticsearch-compatible schema representations and queries.**

We experiment with multiple instruction-tuned base models, including Phi-2 and Mistral-7B-Instruct [3], and evaluate their performance under both zero-shot and LoRA-fine-tuned settings. Our schema-aware prompt design incorporates structured metadata to guide the generation process, ensuring outputs conform to Elasticsearch's strict syntax and semantics.Quantitative and qualitative results show that LoRA-fine-tuned models outperform zero-shot baselines by a wide margin, both in syntactic validity and semantic fidelity. This study demonstrates the effectiveness of lightweight adaptation methods like LoRA when combined with domain-specific prompt engineering and task-specific data construction, opening pathways for robust natural language interfaces to search engines and document databases.

***Index Terms*—LoRA, Elasticsearch DSL, Natural Language Querying ,Query Translation, Low-Rank Fine-Tuning**

I. Introduction

Elasticsearch has become a foundational tool in modern search engines and enterprise-scale data analytics pipelines. It supports complex querying through its powerful Domain-Specific Language (DSL), but writing these queries requires deep familiarity with both the DSL syntax and the underlying schema. This creates a barrier for analysts, business users, and developers who are not proficient with Elasticsearch's query structure.

Recent advancements in large language models (LLMs) have demonstrated their capability to generate structured output from natural language prompts, enabling natural language interfaces to structured systems.This capability is part of a broader, rapidly advancing field where LLMs are being successfully applied to generate code in various programming languages [13].This has been particularly transformative in the domain of database interaction, with a significant body of research focused on translating natural language to SQL [14]. However, full fine-tuning of large-scale LLMs remains computationally expensive and resource-intensive. This limitation has led to growing interest in a family of efficient fine-tuning methods collectively known as Parameter-Efficient Fine-Tuning (PEFT) [6]. Techniques such as Low-Rank Adaptation (LoRA) enable the adaptation of pre-trained models to new tasks by updating only a small subset of trainable parameters.

In this work, we propose a LoRA-based fine-tuning pipeline for translating natural language queries to Elasticsearch DSL. Although we initially experimented with smaller models like TinyLlama-1.1B-Chat and Phi-2, our best results were obtained using Mistral-7B-Instruct, an instruction-tuned LLM. Since no existing dataset supports this specific task, we built a custom dataset of 4,000 examples by adapting SQL-based question sets to the Elasticsearch DSL format and pairing them with schema-aware prompts.

Figure 1 illustrates the full flow of our system. A user provides a natural language query and the associated schema. These are combined into a schema-aware prompt that is passed to a LoRA-fine-tuned LLM. The generated DSL query is then executed in Elasticsearch to return the appropriate results.

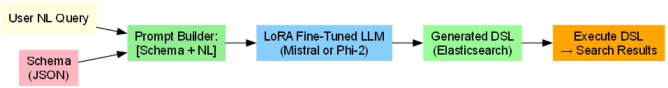

Fig. 1. Overview of Query Generation Pipeline using LoRA Fine-Tuned LLM

Key contributions :

- A novel dataset of 4,000 NL-to-DSL examples based on Elasticsearch schemas and prompts.
- LoRA-based 4-bit fine-tuning of TinyLlama, Phi-2, and Mistral-7B models for efficient query translation.
- Schema-aware prompt engineering that guides the model to generate accurate and valid Elasticsearch DSL queries.
- Empirical results demonstrating significant improvements in syntactic accuracy and semantic fidelity over baseline and zero-shot models.

II. Related Work

Translating natural language to structured queries has gained significant attention with the rise of large language models and their ability to perform zero-shot and few-shot generalization across tasks.

Text2SQL and Structured Query Generation

The Spider dataset [12] introduced a cross-domain benchmark for converting natural language questions into SQL queries. It emphasized schema-awareness and generalization across unseen databases. Models trained on Spider have influenced many follow-up works, including DocSQL and SQLNet. However, most of these efforts focus on SQL rather than NoSQL query languages. While significant progress has been made in converting natural language into SQL for relational databases, limited research exists for search-optimized systems like Elasticsearch, which use their own deeply nested Domain-Specific Language (DSL). This gap is significant, as translating natural language to NoSQL databases presents unique challenges due to varied data models and query languages, a research area that is now beginning to gain more attention [7]

DocSpider: NL to MongoDB Queries

The DocSpider dataset [15] extends Spider by introducing a cross-domain corpus mapping natural language to MongoDB queries. It was generated by translating 10,181 human-annotated Spider NL–SQL pairs to MongoDB Query Language (MQL) using LLMs like GPT-4 and then manually verified. DocSpider contributed a significant training resource with 4,043 training and 620 dev instances, showcasing that large models can generalize to document-style databases. However, it did not explore Elastic DSL, which presents unique challenges due to its nested structure and search-based semantics.

Text2Cypher: Querying Graph Databases

Text2Cypher [9] addressed translating English to Cypher queries for graph databases. The authors created a unified benchmark combining multiple Cypher datasets and fine-tuned LLMs like LLaMA and Gemini. Fine-tuned models achieved significant improvements (BLEU +0.13 to +0.34), showing that schema-aware instruction tuning is crucial. Although Cypher shares structural similarities with SQL, its graph-centric focus differentiates it from Elastic DSL. Although Cypher shares structural similarities with SQL, its graph-centric focus differentiates it from Elastic DSL. The challenge of bridging natural language and structured queries also extends to other database paradigms, such as the semantic web, where researchers have developed frameworks for translating between natural language and SPARQL queries for RDF knowledge bases [5].

Parameter-Efficient Fine-Tuning with LoRA

LoRA [2] is a low-rank adaptation method that enables fine-tuning of only a small number of trainable parameters by injecting adapter layers into attention modules. It reduces memory and compute cost drastically, making it suitable for resource-constrained environments. PEFT libraries have made LoRA-based training accessible for models like TinyLlama and Mistral, facilitating quick adaptation to new domains.

Gap Addressed

Our approach involves schema-aware prompt construction and parameter-efficient fine-tuning techniques to align model outputs with the syntactic and semantic constraints of Elasticsearch. The importance of grounding model outputs in a given schema is a well-recognized challenge in the broader task of generating code from natural language, making it a critical aspect of our methodology [4].

- It focuses on Elasticsearch DSL, a query language with nested filters, keyword matching, and sorting – structures that differ from SQL or MQL.
- It leverages LoRA to fine-tune small LLM's with 4-bit quantization, enabling efficient training on a single-GPU system.

TABLE I
Comparison of Related Work

Work	Target DB	Method	Schema-Aware
Spider [12]	SQL	Fine-tune / Seq2Seq	Yes
DocSpider [15]	MongoDB	LLM + Manual Verify	Yes
Text2Cypher [9]	Neo4j	LLM Fine-tune	Yes
this paper	Elasticsearch	LoRA + Schema Prompt	Yes

Our study adds to this evolving field by extending LLM-based query generation to the Elastic ecosystem and showing that resource-efficient fine-tuning still yields highly usable models.

III. Dataset Design

A crucial aspect of training any natural language interface to structured query systems is the quality and diversity of its training data. To address the lack of large-scale, open datasets for Elasticsearch DSL translation, we curated a custom dataset of 4,000 instances by leveraging the well-established Spider dataset [12] and augmenting it through controlled generation using an LLM.

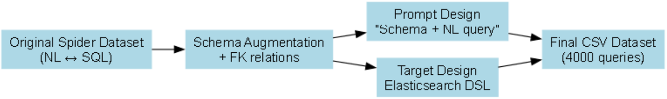

Fig. 2. Preprocessing Pipeline Diagram

Source Dataset: Spider

The Spider dataset was originally created for the task of converting natural language questions to SQL. It spans 200 complex database schemas and over 10,000 manually annotated NL-SQL pairs. Its wide schema coverage and varying query complexity make it an excellent foundation for our translation task, despite being SQL-focused.

Conversion to Elasticsearch DSL

To transform the Spider data into a format suitable for our task, we employed the DeepSeek-V1 language model to generate Elasticsearch-compatible DSL queries from natural language questions. For each example, we structured a prompt containing:

- A concise, schema-aware description listing table names, attributes, and foreign keys.
- A natural language query drawn directly from the Spider dataset.
- An explicit instruction: *"Write only the Elasticsearch DSL with no explanation for the query using the following schema..."*

This prompt ensured the model remained grounded in the schema and produced strictly formatted Elasticsearch syntax without irrelevant commentary.

Example Instance

A representative sample from our dataset:

- **Query:** `How many heads of the departments are older than 56?`
- **Schema:**

```
department(Department_ID, Name,Creation,
Ranking,Budget_in_Billions,
Num_Employees)
head(head_ID, name, born_state, age)
management(department_ID, head_ID,
temporary_acting)
Foreign keys:
management.head_ID = head.head_ID
management.department_ID =
department.Department_ID
```

- **Target Elasticsearch DSL:**

```
GET /head/_count {
  "query": {
    "range": {
      "age": { "gt": 56 }
    }
  }
}
```

Intuition Behind Dataset Creation

Translating natural language to Elasticsearch DSL poses unique challenges that differ from SQL. Elastic queries are nested, lack standard joins, and are highly contextual in search-based operations like keyword filtering, range scans, and aggregations.

By generating the DSL from natural language via LLMs under schema constraints, we ensure that:

- The model is trained on syntactically valid, schema-grounded queries.
- It learns to map natural language intent to the structural patterns of Elasticsearch (e.g., `match`, `range`, `terms`, `bool` filters).
- Domain adaptation is feasible with fewer examples due to consistent prompt format and schema guidance.

This technique of leveraging a powerful LLM to create a new, instruction-based dataset aligns with established methodologies for model alignment and data synthesis [11].

Model Trials and Final Selection

We initially experimented with several lightweight instruction-tuned language models including TinyLlama-1.1B-Chat and Microsoft's Phi-2. These models were attractive due to their small memory footprint and fast inference capabilities, especially on consumer-grade GPUs. However, during early testing, their performance in translating natural language to Elasticsearch DSL was inconsistent. Specifically, we observed issues such as incomplete query generation, incorrect nesting, and failure to recognize key schema elements. As a result, we transitioned to using `mistralai/Mistral-7B-Instruct-v0.1`, an instruction-tuned model known for its strong performance on multi-turn question answering and code generation tasks. Such models are specifically fine-tuned to better understand and follow user commands, a critical capability for generating syntactically precise DSL queries [8]. To enable efficient training and inference despite the increased model size, we applied 4-bit quantization using techniques from QLoRA [1], which allow large models to be fine-tuned on limited hardware without significant performance degradation. Mistral yielded superior results in terms of DSL structure accuracy and semantic fidelity.

Training Results and Inference

Throughout the training process, we observed a consistent and substantial reduction in training loss. The model rapidly learned to align the natural language query with the correct Elasticsearch DSL output. Early in training, losses fell sharply, and later epochs showed stabilization, indicating convergence. By the end of training, the model consistently produced syntactically valid and semantically correct queries across a range of test inputs.

The choice of instruction-tuned Mistral, combined with LoRA and 4-bit quantization, proved to be a powerful strategy for resource-efficient domain adaptation. The model demonstrated high fidelity in understanding both schema context and user intent, making it highly suitable for integration into real-world applications where users interact with structured Elasticsearch indices through natural language interfaces.

IV. Results and Analysis:Mistral-7B

To evaluate the performance of the fine-tuned model, we compared its generated Elasticsearch DSL outputs with ground truth DSL queries using a custom similarity metric.The choice of a custom, structurally-aware metric was motivated by the limitations of standard text-based metrics for code, a challenge addressed by specialized metrics like CodeBLEU [10] which consider syntactic and data-flow properties. This metric is based on structural JSON similarity using ‘SequenceMatcher‘ after normalization. The analysis was conducted on a held-out test set, and we compared results against the base (unfine-tuned) Mistral-7B-Instruct and Microsoft-Phi-2 models.

Quantitative Metrics

Our LoRA fine-tuned model achieved:

- **Mean Similarity Score:** 0.7131
- **Median Similarity Score:** 0.7155
- **Mean Base Model Similarity:** 0.1788
- **Average Improvement over Base:** 0.5343
- **Improved Queries:** 80
- **Worsened Queries:** 2
- **Unchanged Queries:** 0

Similarity Score Distribution

As illustrated in Fig. 3, the base model’s similarity scores (in red) are heavily skewed toward lower values, with many queries producing nearly zero overlap with the target DSL. In contrast, the LoRA model (in green) demonstrates a broader and more right-skewed distribution, with many queries achieving scores near or above 0.8. This indicates a significant enhancement in output structure and accuracy after fine-tuning.

LoRA vs. Base Comparison

Fig. 4 presents a scatter plot of LoRA similarity scores versus base model scores. The majority of points lie above the $y = x$ line, confirming that the LoRA model improves over the base model in almost all cases. The steep rise in similarity values also showcases the effectiveness of schema-aware prompting and LoRA adaptation in guiding structured generation.

Boxplot Summary

The boxplot in Fig. 5 summarizes score distributions. The base model has a much lower interquartile range (IQR), centered around 0.1–0.2, while the LoRA model demonstrates a higher median and tighter spread toward the upper end of the scale. Outliers in the base model reflect unpredictable or invalid generations, while the LoRA model maintains stability across prompts.

Overall Insights

These evaluation results confirm that LoRA fine-tuning on a structurally rich, schema-augmented dataset significantly improves the model’s ability to translate natural language into semantically and syntactically accurate Elasticsearch DSL. The consistent improvement across diverse queries indicates the model’s generalization capabilities and robustness.

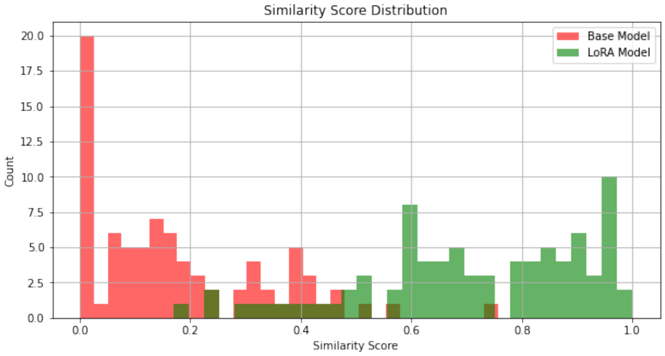

Fig. 3. Histogram of Similarity Score Distribution

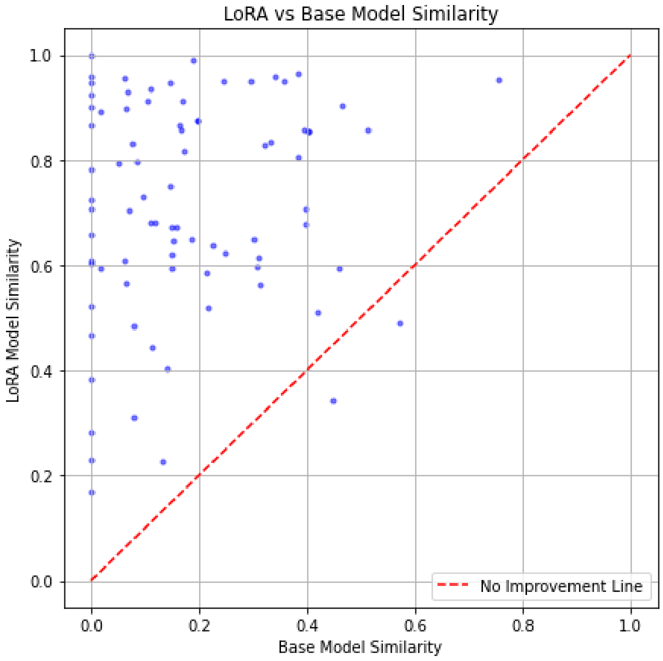

Fig. 4. Scatter Plot: LoRA vs. Base Model Similarity

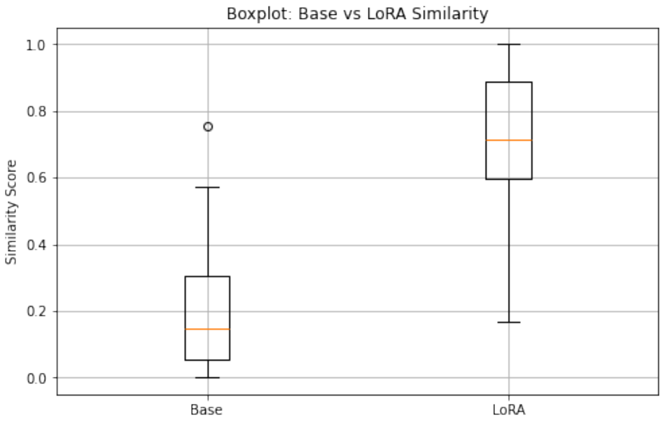

Fig. 5. Boxplot: Base vs. LoRA Model Similarity

V. Results and Analysis: Phi-2

To evaluate the effectiveness of LoRA fine-tuning on smaller instruction-aligned models, we applied the same training pipeline to Microsoft's `Phi-2` model. Phi-2 is a compact language model designed for efficient inference while retaining strong instruction-following capabilities. We used the same 4,000-instance dataset, schema-aware prompting strategy, and LoRA configuration as with Mistral to ensure consistency across experiments.

Quantitative Metrics

The LoRA fine-tuned Phi-2 model achieved the following results:

- **Mean Base Model Similarity:** 0.1968
- **Mean LoRA Model Similarity:** 0.5271
- **Average Improvement:** 0.3303
- **Improved Queries:** 229
- **Worsened Queries:** 35
- **Unchanged Queries:** 0

These results show that while Phi-2 benefits from fine-tuning, its smaller architecture and limited capacity may constrain performance in generating deeply structured DSL queries.

Score Distribution

Fig. 6 presents the histogram of similarity scores for the Phi-2 model before and after LoRA fine-tuning. The base model scores are heavily concentrated in the lower range (below 0.3), with the LoRA-finetuned model shifting the distribution toward moderate similarity values (0.4–0.6) and a smaller number of high-quality predictions above 0.7.

Improvement Visualization

The scatter plot in Fig. 7 provides a point-wise comparison between base and fine-tuned predictions. Most data points are located above the $y = x$ diagonal, indicating that LoRA tuning led to consistent performance improvements across the majority of examples, although some outliers exhibit limited or negative gain.

Interpretation

Phi-2, due to its smaller parameter count, serves as a cost-effective alternative for query translation in resource-constrained settings. Despite not reaching Mistral-level performance, its post-tuning gains demonstrate that lightweight models can still adapt well to DSL generation tasks when guided with schema-augmented prompts and structurally rich supervision. For applications requiring high throughput on limited hardware, Phi-2 remains a viable option.

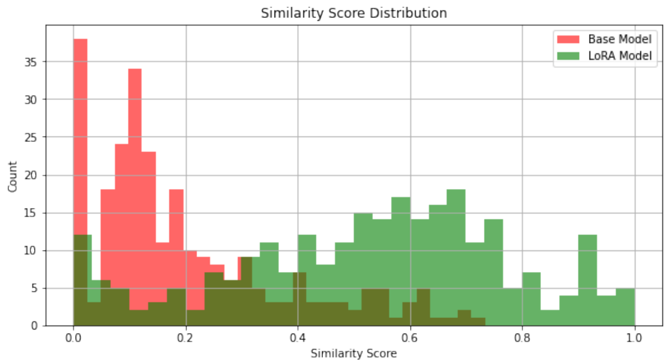

Fig. 6. Histogram of Similarity Score Distribution for Phi-2

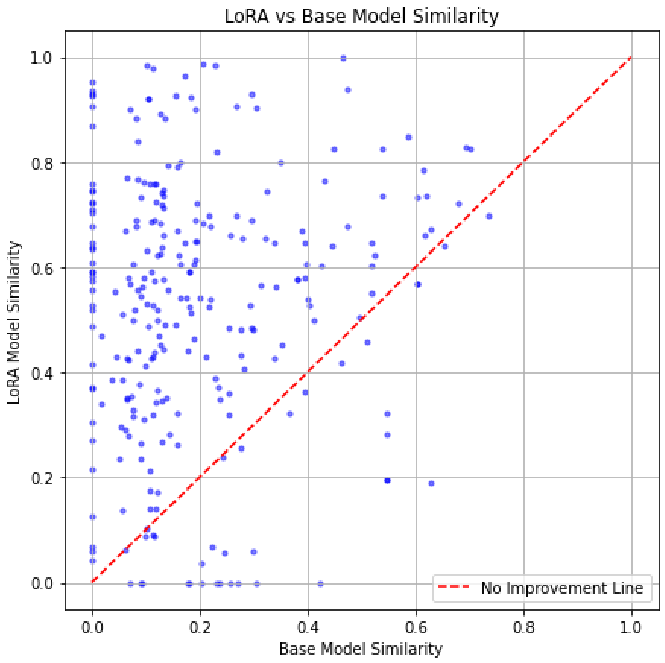

Fig. 7. Scatter Plot: LoRA vs. Base Model Similarity for Phi-2

References

[1] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2023. Submitted on 23 May 2023.

[2] Edward Hu, Yelong Shen, Phil Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022.

[3] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lafmple, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *Mistral AI Blog*, 2023. Available at https://mistral.ai/news/announcing-mistral-7b/.

[4] Juyong Jiang, Fan Wang, Jiasi Shen, Sungju Kim, and Sunghun Kim. A survey on large language models for code generation, 2024.

[5] Gwénolé Lecorvé, Morgan Veyret, Quentin Brabant, and Lina M. Rojas Barahona. SPARQL-to-text question generation for knowledge-based conversational applications. In Yulan He, Heng Ji, Sujian Li, Yang Liu, and Chua-Hui Chang, editors, *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 131–147, Online only, November 2022. Association for Computational Linguistics.

[6] Vladislav Lialin, Vijeta Deshpande, Xiaowei Yao, and Anna Rumshisky. Scaling down to scale up: A guide to parameter-efficient fine-tuning, 2024.

[7] Jinwei Lu, Yuanfeng Song, Zhiqian Qin, Haodi Zhang, Chen Zhang, and Raymond Chi-Wing Wong. Bridging the gap: Enabling natural language

queries for nosql databases through text-to-nosql translation, 2025.

[8] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022.

[9] Makbule Gulcin Ozsoy, Leila Messallem, Jon Besga, and Gianandrea Minneci. Text2cypher: Bridging natural language and graph databases. *arXiv preprint arXiv:2412.07867*, 2024. Submitted on 13 Dec 2024.

[10] Shuo Ren, Daya Guo, Shuai Lu, Long Zhou, Shujie Liu, Duyu Tang, Neel Sundaresan, Ming Zhou, Ambrosio Blanco, and Shuai Ma. Codebleu: a method for automatic evaluation of code synthesis, 2020.

[11] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions, 2023.

[12] Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir R. Radev. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2018.

[13] Zibin Zheng, Kaiwen Ning, Yanlin Wang, Jingwen Zhang, Dewu Zheng, Mingxi Ye, and Jiachi Chen. A survey of large language models for code: Evolution, benchmarking, and future trends, 2024.

[14] Xiaohu Zhu, Qian Li, Lizhen Cui, and Yongkang Liu. Large language model enhanced text-to-sql generation: A survey, 2024.

[15] Arif Görkem Özer, Recep Firat Cekinel, Ismail Hakki Toroslu, and Pinar Karagoz. Docspider: A dataset of cross-domain natural language querying for mongodb. In *Proceedings of the 2024 International Conference on Information and Knowledge Management (CIKM)*. ACM, 2024.