# Text to SQL: For A Custom Query Language (SIEM)

**Shashank Nigam**

shashanknigam40@yahoo.com

## Abstract

Text to SQL is a Natural Language Understanding task which translates grounded questions against a schematized database into an executable query language. The task has gained traction over the recent years with different benchmark databases such as WikiSQL, SPIDER and BIRD bench were created to solve the problem. The present state of the art auto regressive large language models such as GPT-3.5, GPT-4, CLAUDE and its variants have been able outperform most of these benchmarks, indicating ability of these models understand underlying data structure, table relationship and complexities of SQL query. While structured query language model is useful in most of the domains, some domains such as Cyber-Security operate on unstructured data. Security Incident and Event Management (SIEM) is a query language used by security analyst for threat detection and compliance management. SIEM tools are designed to ingest log format data from devices, networks, and application managed. SIEM tools have a structure similar to a SQL, however they operate over unstructured collections of data. The functions defined in this custom query languages do not directly map to SQL functions. Language models which are optimized to generate responses for Structured Query Language tend not to perform well on these custom query language model. We explore the application of pipeline based method for extending current state of the art text to SQL models to text to custom SIEM generator.

## 1 Introduction

Text to SQL parsers (Zhong et al., 2017), (Xu et al., 2017), is a natural language understanding task which translates grounded questions into an executable structured query language (SQL). The task allows non data analyst user to extract relevant information from the databases. Some of the prominent datasets which facilitated in advancement of the field include WikiSQL (Zhong et al., 2017), Spider (Yu et al., 2018) facilitated the sequence to sequence models such as T5 (Raffel et al., 2023) to act as an interface for Text to SQL parsing. Models trained on these dataset understand the complex structure of the language. With the recent progress in large language models there has been a significant advancement in this field. The auto regressive models have achieved state of the art performance in SPIDER (Yu et al., 2018) and WikiSQL (Zhong et al., 2017). Without enough context, model are vulnerable to adverse perturbation (Pi et al., 2022), indicating that the model rely on knowing the underlying schema structure for accurate prediction. The performance of the model further decreases when operating on large number of tables, context compression when tables have large and noisy data sizes as indicated by (Li et al., 2024). Performance of Large language model on these adversaries can be improved by providing domain specific information as indicated by (Wang et al., 2024).

Security Incident and Event Management (SIEM) (SIE, 2024), is a business tool to respond to threats before they affect the business operation. SIEM tools extend the functionality of the SQL on unstructured datasets. SIEM tools ingest device, network and application data from different sources. SIEM tools operate mostly on unstructured dataset, where relation between the source data is not defined. SIEM instruct language model are proprietary e.g. Kusto Sequel Language KQL (microsoft, 2024a) or Splunk Query Language (splunk inc, 2024). Based on the tools there are language specific custom or built in functions defined for them (Inc, 2024). There might be one to one or one to many or no mapping between these custom functions and SQL functions as illustrated in table 2 While Text to SQL models might understand the relation between question and table structure they might not be directly extended for generation of custom SIEM instruct command based on data pro-

Table 1: Function Comparison

| SPL Function | SQL Equivalent | Exact Map |
|---|---|---|
| round() | round() | yes |
| anomaly() | Not Present | no |

vided without further fine tuning. With a suitable pipeline the current state of art text to sql models can be extended for generation of custom SIEM query language. Given a collection schema and a SQL equivalent form of SIEM query a text to SQL model can be fine tuned to generate domain specific SQL query for SIEM tools. A further downstream sequence to sequence model can be used to translate SQL to a custom language model. While a text to SQL model provides a good understanding data relations, a sequence to sequence model can perform translation of from one instruct language to an other. This is illustrated in 1

We used the existing BIRD (Li et al., 2024) dataset with an approach of Few Shot QA approach to generate SQL responses. A chain of thought approach is used for generation of query, with database schema and context with respect to the question asked is provided as the context for the prompt. The response is evaluated for its execution accuracy, which helps in optimizing the query prompts. The responses are then fed to a sequence to sequence model (T5-base) to generate response for custom SIEM tool (shashank, 2024). In addition to the BIRD dataset, handcrafted dataset from tradition SIEM tools is also appended to the dataset. This provides domain specific information azure infrastructure resources making the dataset more diverse. It was observed that the efficiency of downstream code translator (T5-base) dependent on the token size and hence the model performed better on small queries. The approach was trained on a small subset of the dataset and it can be further extended to generate a more complex structure.

## 2 Related Work

Custom SIEM query generation has been incorporated by some of the industry players namely Splunk and Microsoft. Building AI assistant for Splunk discusses the approach for building domain specific search processing language given a natural text (Vialard, 2023). It uses a Sequential transformer based architecture to translate natural language into SPL. Dataset used for training the model includes 1707 high quality data pairs curated by cloud sourcing, Crafted dataset by translating SQL component from the dataset such as WikiSQL (Zhong et al., 2017) and Spider (Yu et al., 2018) with an help of internal Splunk compiler. The model was trained on t5-small transformer based architecture (Raffel et al., 2023) to build this translation system. The model however faces a few challenges as discussed

- Unable to detect underlying table name for the given query. Author attributes this to training data on WikiSQL (Zhong et al., 2017)
- Large query size might generate incorrect results.
- Limited training data, reduced the model's capability to learn the relation among different functions.

Recently, large language models have improved the performance on most of the Text to SQL tasks (Li et al., 2024). The improvements can be indicated by the performance improvements in the State of the art benchmarks on these datasets. The model relies on the context information provided as part of the prompt. The model's performance thus reduces when there is a significant growth in the dataset, or database size. This requires having large knowledge reasoning and context compression for such models to generate accurate results. Benchmarks such as BIRD (Li et al., 2024) explores the impact of having large data schema and database on the model performance.

Performance of the model can be improved by providing more domain specific context while prompting (Wang et al., 2024) leading to construction of intelligent agents operating and specializing on specialized domains. Large language models, outperforms on most of the Text to SQL tasks, fine tuning for specific query language is expensive and difficult to deploy. Domain specific models can be extracted into much smaller models e.g. SQL-Llama (7B) along with instruct agent dataset.

## 3 Data

The proposed system consists of two models, a large language model section which optimizes text to SQL generation and sequential model which relies on SQL to SIEM code generation.

1. Text to SQL dataset: The data consists of data extracted from BIRD-Bench. The process of construction of individual dataset is illustrated 2.

A text to SQL dataset consists of two components Database schema information. Illustrated as below

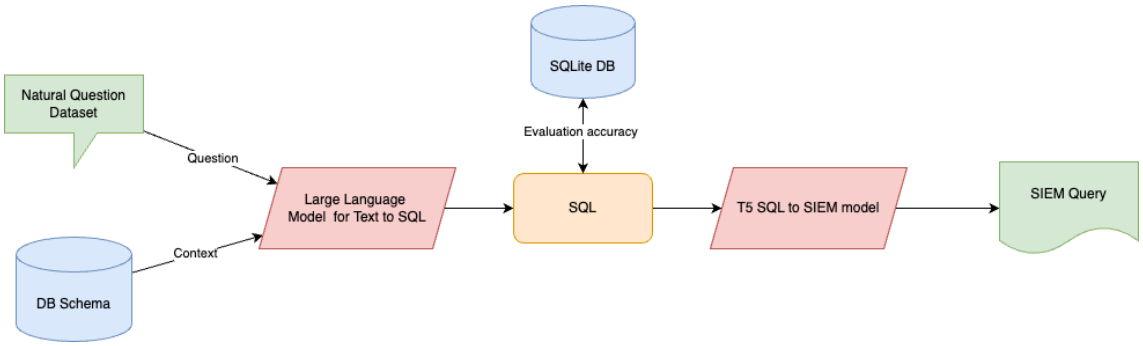


Figure 1: workflow.

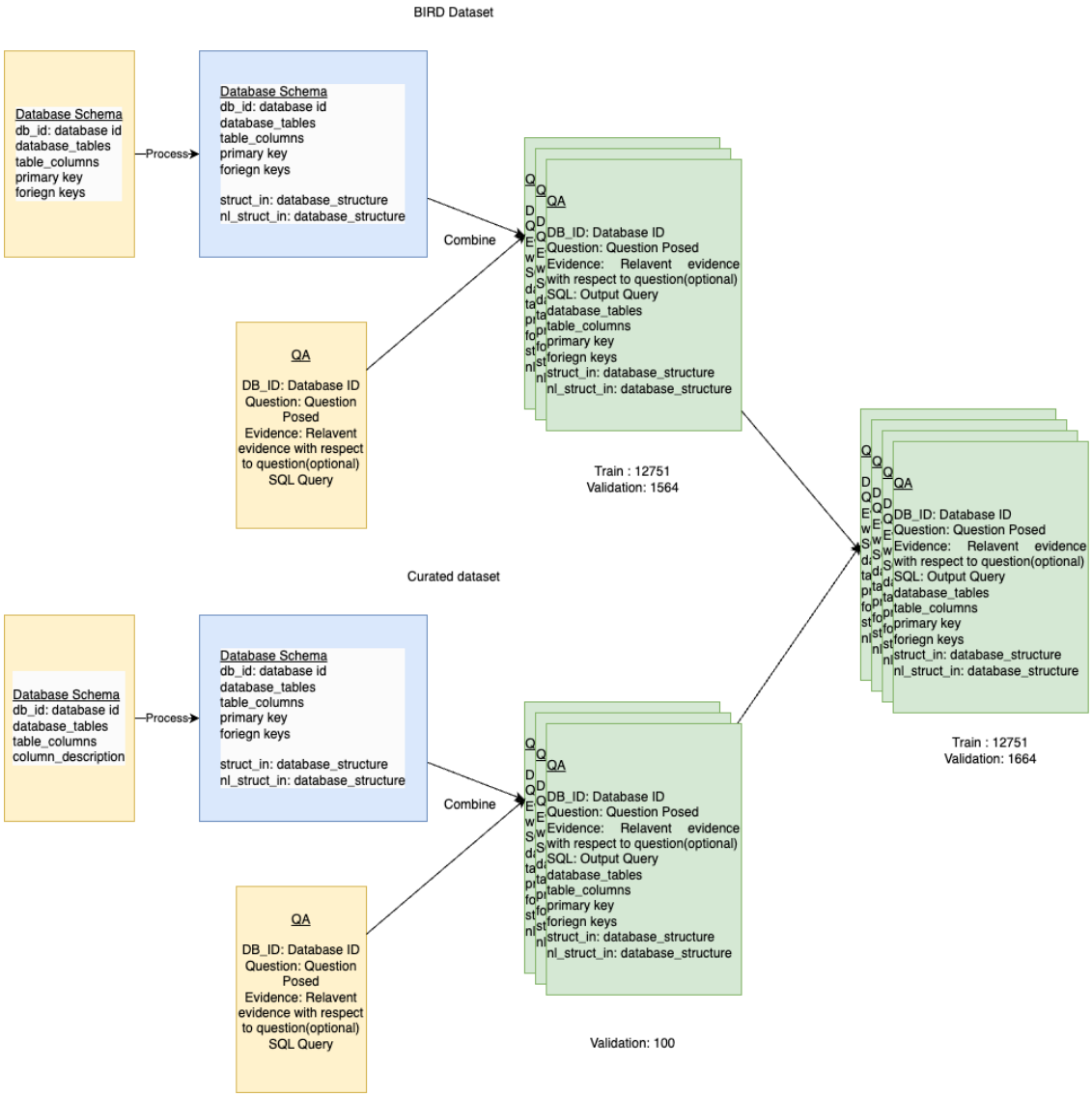


Figure 2: workflow.

- $db\_id$: Database Name
- $db\_tables$: List of tables in the database
- $db\_columns$: List of columns in the for each table in the database
- $db\_column\_type$: Datatype of each column in the dataset.
- $primary\_key$: Primary key for the table
- $foriegn\_key$: Foreign Key relation among the tables
- $struct\_in$: Calculated field which stores all

tables and column names in string format. If does not contain any information about primary key or foreign key relations
- $nl\_struct\_in$: Calculated field which stores all tables and column names in natural language sentence format. It also stores primary key and foreign key relationships

A typical question answer dataset contains below information

- $db\_id$ Database Name
- $query$: Natural language question
- $evidence$: Natural language context provided in addition to the current question
- $SQL$: Expected SQL text for the given natural language context

Bird Consists of 12791 training data and 1584 validation data across 95 different datasets. In addition, to test the cyber security domain additional dataset is curated. We make use of azure infrastructure data scraped from (microsoft, 2024b) consisting of 451 tables each stored as a different database. Queries with respect to this dataset is sourced from (reprise99, 2024) and (microsoft, 2024c). The dataset consists of 12571 in train set and 1694 in validation set.

2. SQL to SIEM model: The dataset consist of 100 hand curated dataset for sql to SIEM translation, where $text\_in$ represents the input SQL text and $text\_out$ represents the output SIEM query expected

## 4 Model

To evaluate the effectiveness of the proposed system we follow a two step approach as illustrated in 1.

1. Text to SQL approach: We compare the performance of the auto-regressive Large Language model namely GPT 3.5 for text to SQL generation. A Chain of Thought Approach is used to enhance the prompt on an evaluation metric of exact match. For in context learning a combination of $struct_in$ indicating the database structure, $evidence$, $nl_struct_in$ is provided as an evidence to the model

2. SQL to SIEM model generation: A T5 base model was trained on BIRD's train dataset for 10 epochs optimized with Adam with learning rate of 0.0001. The model was further finetuned for SQL to SIEM generation for example size of 100.

## 5 Experiments

The overall task was divided into two fields. 1. Text to SQL generation:

DSPy (Khattab et al., 2023) library was used to evaluate GPT-3.5 for Text to SQL generation. The model was evaluated with approaches for zero-shot QA, where only question was provided as the prompt. Few Shot QA. The library used evidence, $struct\_in$, and $nl_struct_in$ as the context and few shot iterations were conducted for a maximum hops of 3. Exact match was used to optimize the DSPy module for Text to SQL generation. The model was used to generate predicted SQL output text. The generated SQL was further evaluated for execution accuracy against BIRD's evaluation metrics (Li et al., 2024). Custom generated queries were excluded from this evaluation

Table **??** summarizes the results from the model evaluation

Table 2: Effectiveness of the Model

| Few-shot | Execution Accuracy GPT-3.5 |
|---|---|
| 0-shot | 24.6 |
| 1-shot | 40.2 |
| 2-shot | 45.2 |

2. SQL to SIEM query generation: A T5-base model was first trained on BIRD-BENCH (Li et al., 2024) development dataset. The model was trained with sequence $seq\_in =< database_name > | < question > | < database\_structure$ where $< database\_structure >= db\_id : Database\_Name|table\_name :< list\_of\_columns >$ and $text\_out = db\_id|SQL$ is the input output text pairs for which Base T5 model was trained for 10 epochs with scoring of exact match. The exact match score after 10 epochs 3

Table 3: Effectiveness of T5-base Model

| Epochs | Exact Match |
|---|---|
| 10 | 41.56 |

The model was further finetuned on the task of SQL to SIEM (shashank, 2024) for 10 epochs on evaluation metrics of exact match. 4

The finetuning was conducted on a very limited dataset and can not be generalized for a large sample size

Table 4: Effectiveness of T5-base Model

| Epochs | Exact Match |
|---|---|
| 10 | 62.56 |

## 6 Analysis

The objective of the experimentation was to test whether the existing state of the art pipeline for Text to SQL can be extended to custom SIEM generation. A pipeline based approach was used for the same. Based on the experiments conducted, Text to SQL model performed with auto regressive backend (GPT-3.5) when context was provided. The models performance increases when table structure as $struct\_in$ is provided. Some additional information such as primary key, foreign key relations would help the model to perform reason better. A chain of thoughts experimentation was used for this evaluation. Increasing number of hops with chaining boosted the performance of the model. However, the custom data passed to the model as part of the validation test did not do well with the optimization process. The schema generally consisted of single table with large description, which increased the context size for the model. Summarizing of the context in addition might help to boost the accuracy of the model on validation set which needs to be evaluated.

T5-base model performance was evaluated on two task of text to SQL and SQL to SIEM. The limited data size for SIEM was the rationale behind this approach. It was observed, that model performance improved with each epoch. Model is consistent in generating coherent response for the provided SQL text. The evaluation data size is small and not complex (did not include any join condition), hence model over-fitting might be a possible reason. T5-base model generates coherent SQL response for simple query with query length less than 100 character. The model is able to detect database name in addition of generating query. Limited response from the model can be attributed to limited token space for the T5-base model. The downstream model needs to be further evaluated on large model.

## 7 Conclusion

For a small sample size the hypothesis of extending text to SQL to generic text to custom SIEM was evaluated for this paper. While current experimentation provides a average result on custom SIEM model. For text to SQL models still rely on underlying schema structure for efficient query generation. Prompt engineering where the details of the schema can be masked needs to be further evaluated. As a future scope of work, would be to increase the data size and make examples more complex. Further other standard models such as GPT-4, Llama-3 etc needs to be evaluated on a similar task

## References


2024. Siem.

Splunk Inc. 2024. spl2 search manual.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. 2023. Dspy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*.

Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Ruiying Geng, Nan Huo, et al. 2024. Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls. *Advances in Neural Information Processing Systems*, 36.

microsoft. 2024a. Kusto query language.

microsoft. 2024b. msschema.

microsoft. 2024c. sentinalqueries.

Xinyu Pi, Bing Wang, Yan Gao, Jiaqi Guo, Zhoujun Li, and Jian-Guang Lou. 2022. Towards robustness of text-to-SQL models against natural and realistic adversarial table perturbation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2007–2022, Dublin, Ireland. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer.

reprise99. 2024. reprise99.

shashank. 2024. customsiem.

splunk inc. 2024. Splunk processing language.

Julien Veron Vialard. 2023. Building an 'ai assistant' for splunk.

Bing Wang, Changyu Ren, Jian Yang, Xinnian Liang, Jiaqi Bai, Linzheng Chai, Zhao Yan, Qian-Wen Zhang, Di Yin, Xing Sun, and Zhoujun Li. 2024. Mac-sql: A multi-agent collaborative framework for text-to-sql.

Xiaojun Xu, Chang Liu, and Dawn Song. 2017. Sqlnet: Generating structured queries from natural language without reinforcement learning.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018. 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 Process-*

*ing*, Brussels, Belgium. Association for Computational Linguistics.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2sql: Generating structured queries from natural language using reinforcement learning. *CoRR*, abs/1709.00103.