NL-Based Database Administration for Handling Heterogeneous Datasets Using Finetuned LLM

Pradnya Sawant[a] and Kavita Sonawane[b]
St. Francis Institute of Technology, Mumbai, India
fi

Keywords: Database Administrator Queries, Data Definition Language, Data Manipulation Language, Data Control Language, Large Language Model, NLIDB.

Abstract: Translating natural language (NL) questions into structured query language (SQL) queries is becoming increasingly important for making databases easier to use and manage. Different large language models (LLMs) have been used for this translation in recent years. These models are mostly trained and evaluated on datasets covering a few types of data manipulation language(DML) queries like projection, selection, aggregate functions, joins, etc. However, these datasets failed to contain queries required for Database Administrator(DBA) operations such as creating and modifying database schema, managing user permissions, etc. This paper presents an approach to help database administrators (DBAs) and end users interact with databases more intuitively by generating SQL queries from natural language inputs. As no such dataset is publicly available, we have created a specialized dataset called DBASQL, which includes common DBA operations addressing data definition language(DDL), data manipulation language(DML), and data control language(DCL) related natural language questions like creating tables, views, or indexes; inserting values; updating data types or values; renaming tables or columns; granting or revoking user permissions, paired with their corresponding SQL queries. For experimentation, we have finetuned Text-to-Text Transfer Transformer (T5) Large on our customized DBASQL dataset, aiming to improve the accuracy of these translations. Our evaluation shows that this approach effectively translates NL to SQL that addresses DBA operations, making it easier to handle DDL, DML, and DCL database operations without requiring extensive SQL knowledge. This research highlights the potential of NLP models to improve the efficiency of natural language to SQL translation by enabling smarter database interfaces for DBA as well. Also, the proposed DBASQL dataset can be integrated with any heterogeneous datasets, such as single-domain and cross-domain, for the translation of natural language to SQL queries. Hence, covering the border range of SQL queries that can be used by both end users and database administrators.

1 INTRODUCTION

With data growing quickly, managing databases efficiently is more important than ever. However, accessing database information often requires knowing query languages like SQL, which can be challenging for non-technical users.

Natural Language Processing (NLP) provides a solution by enabling non-technical users to ask questions in natural language rather than knowing complex SQL commands. NLP models convert these natural language questions into appropriate SQL queries, making it easier for both Database Administrators(DBA) and non-technical users to work with databases.

Current NL to SQL systems have shown success in general-purpose applications, but adapting these models for DBA-specific operations presents unique challenges. The systems already available have improved a lot in converting natural language into SQL, but most datasets mainly focus on a few Data Manipulation Language (DML) queries. On the other hand, queries for Data Definition Language (DDL), such as CREATE and ALTER, Data Manipulation Language(DML) like CREATE, INSERT, UPDATE, and Data Control Language (DCL), like GRANT and REVOKE, are not well-represented. DBA-specific natural language questions involve complex queries to modify schema structures, update data types and values, grant and revoke permissions, etc., that require precise SQL commands tailored to each database's

[a] https://orcid.org/0000-0003-3982-4077
[b] https://orcid.org/0000-0003-0865-6760

Sawant, P. and Sonawane, K.
NL-Based Database Administration for Handling Heterogeneous Datasets Using Finetuned LLM.
DOI: 10.5220/0013608700004664

In *Proceedings of the 3rd International Conference on Futuristic Technology (INCOFT 2025) - Volume 3*, pages 56-64
ISBN: 978-989-758-763-4

configuration. This paper addresses these challenges by fine-tuning a recent large language model (LLM), T5-large, specifically for DBA-related SQL queries. As there is no such dataset available publicly for handling all types of DBA queries, we created a customized ”DBASQL” (Database Administrator Structured Query Language) dataset containing natural language queries commonly used by database administrators. This dataset includes natural language questions such as “Modify datatype of custid column from integer to number”, “Allow insert operation on user table to the manager”, and “Create table employee having id, name, age”. These questions are paired with corresponding SQL commands, forming a robust dataset for training and fine-tuning advanced LLMs to achieve accurate and reliable translation. This dataset can be used in combination with other existing heterogeneous datasets, like single-domain and cross-domain to cover the range of SQL operations through natural language questions. The proposed DBASQL dataset is publicly available at https://www.kaggle.com/datasets/pradnyasawant/dbasql. Also, the proposed model can be used to handle complex tasks like referencing schemas and updating database content and table schema, and managing user permissions on database objects.

Through experiments, we show that our model performs with high accuracy, highlighting the potential of T5 Large in automating database administration tasks. By minimizing the need for deep SQL knowledge, this research helps create a smarter Natural language Interface to Database (NLIDB) for managing DBA operations as well.

2 RELATED WORK

Converting natural language (NL) queries into SQL has been studied for a long time. Early methods relied on rule-based systems, but these were limited because they were rigid and couldn’t handle a wide variety of queries or complex database structures cite Kumar14. Neural network-based models brought big improvements to converting natural language into SQL, especially with sequence-to-sequence (Seq2Seq) models. For example, Seq2SQL used reinforcement learning to create SQL commands, solving issues with query structure and accuracy (V. Zhong and Socher, 2017). Later models like SyntaxSQLNet (T. Yu, 2018b), F-SemtoSql (Q. Li and Zhong, 2020) and TypeSQL (T. Yu and Radev, 2018) became even more accurate by adding rules and using information about data types. They also introduced attention mechanisms, which helped the models focus on the important parts of the input, making it easier to handle more complicated queries.
Along with NLIDB, NL2VIS(Natural Language to Visualizations) systems like NL4DV(A. Narechania and Stasko, 2021), Advisor(C. Liu and Yuan, 2021), ncNet(Y. Luo and Qin, 2022)are becoming popular as non-technical users can generate business insights using charts, graphs, etc. from the underlying database. There are different benchmarks available for generating visualizations through natural language questions (K. Z. Hu and et al., 2019)(Y. Luo and Qin, 2021).

The Transformer architecture completely changed NLP-to-SQL tasks, with models such as BERT(J. Devlin and Toutanova, 2018), RoBERTa(K. Ahkouk and Ennaji, 2021), XLNet(Q. Li and Zhong, 2020), T5(Y. Li and Zhang, 2023), and Codex(Trummer, 2022) providing better context understanding, which improved performance in generating SQL queries. Fine-tuning models like T5 on SQL datasets made NL-to-SQL translation more reliable and adaptable, while OpenAI’s Codex model showed great ability in generating SQL commands across many different types of queries. (T. Yu, 2018a) introduced a large, cross-domain dataset with complex multi-table SQL queries, which became an important benchmark for evaluating recent large language models(LLMs)(M. A. Khan and Azam, 2024)(N. T. K. Le and Teshebaev, 2023)(C. Raffel and Liu, 2020). This dataset has driven the development of advanced models that can be generalized effectively. The Spider dataset is a popular benchmark in NLP-to-SQL research. It contains natural language questions linked to complex SQL queries for many different database schemas. Spider is great for testing how well models can work with new database structures and handle multi-table joins and nested queries. However, it mainly covers DML queries and does not include database administration-related DDL, DML, and DCL queries. WikiSQL is another well-known dataset that is simpler to use. It focuses on single-table queries created from Wikipedia tables. However, since it only includes SELECT queries and doesn’t cover the DDL and DCL queries (T. Yu, 2018a).

CoSQL (T. Yu and Su, 2019a)and SParC(T. Yu and Su, 2019b) are based on the Spider dataset, but add conversational and multi-step queries, where users improve their queries gradually. However, like Spider, they mostly focus on DML queries and do not cover DDL or DCL queries in detail. While current NLP-to-SQL datasets provide a strong foundation for developing models that handle DML queries, there is a clear gap in datasets representing DBA-related DDL and DCL queries. Addressing this gap is important for developing a robust NL-to-SQL system that can handle all database management queries.

Even with these advancements, applying models to database administration (DBA) operations is an area of growing research. DBA operations, like managing permissions, manipulating structures of schema, changing database contents, and making complex schema references, which general-purpose models struggle to handle. Models like RAT-SQL (B. Wang and Richardson, 2020) and SmBoP (Z. Zhao and Liang, 2021) focus on schema encoding and semantic parsing of a few DML queries, but they do not address queries related to DBA operations. Hence there is a need to fine-tune LLM for DBA operations, which help better manage these administration operations, creating a more user-friendly interface for DBAs with limited SQL knowledge. This paper builds on these advancements by fine-tuning the T5 large model and developing a specialized dataset for schema and database content manipulation, handling user permissions, etc. Our work demonstrates the model's capability in generating accurate SQL queries for DBA needs, broadening NLP-to-SQL applications, and addressing gaps in handling specialized DBA operations, ultimately supporting the development of smarter, more accessible NLIDB.

3 PROPOSED ARCHITECTURE

Figure 1 shows the proposed architecture for NL-based database administration. In this, a natural language question addressing DBA queries is first preprocessed using tokenization, padding, etc. Then the NLP module will be responsible for converting this preprocessed text into the appropriate SQL query. The highlighted part in the figure shows that we have fine-tuned the T5 large model for NL processing specifically for DBA-related queries. This generated SQL query is executed against the relational database through which the output can be produced. For Fine-Tuning the T5 large model, we have not considered the table schema as input which can reduce the complexity of the model. The finetuning of the T5 model is explained in Section 3.1.

3.1 Finetuning of T5 Large

The model is initialized with pre-trained weights from T5-large, and the input is formatted to include the natural language questions. The output is the target SQL query. Both input and output are tokenized to fit the model vocabulary, with careful attention to sequence truncation and padding. The loss is calculated on the generated SQL sequences compared to the ground truth. By optimizing the model using techniques like learning rate scheduling and validation on held-out examples, T5-large learns to generalize its translation capabilities, even for complex SQL queries. For training the model we have used our own customized ”DBASQL” dataset, which is explained in detail in Section 3.2.

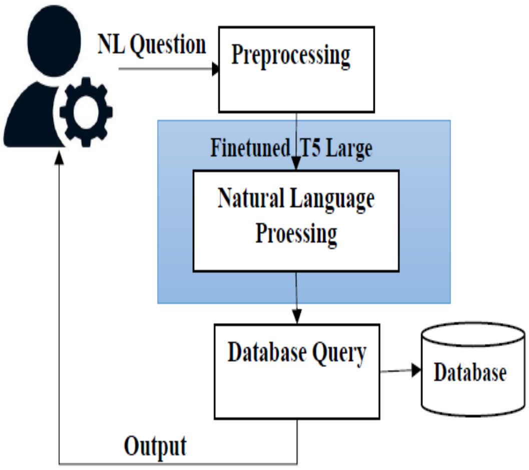


Figure 1: Proposed Architecture for NL-based Database Administration.

3.2 Proposed Dataset: DBASQL Dataset

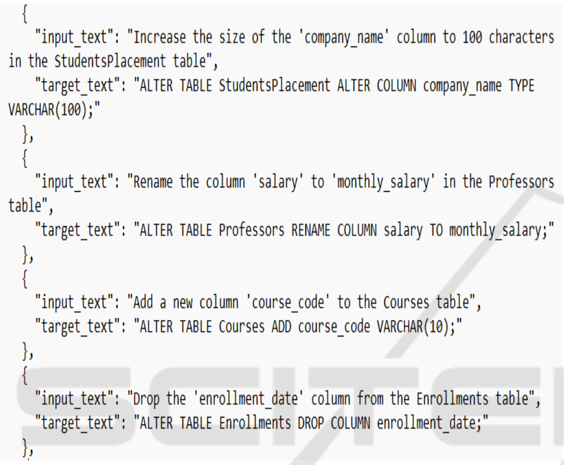
There is no publicly available dataset specifically designed for NL to SQL translation that comprehensively handles Data Definition. Language(DDL), Data Manipulation Language(DML), and Data Control Language(DCL) queries for database administration (DBA) requirements. Most existing datasets, such as Spider, focus on SQL query generation for various databases but tend to emphasize SELECT statements (DML) and not the full range of DBA-related tasks, such as creating, altering, or managing database structures (DDL). Developing a dataset tailored to DBA requirements would require incorporating queries that handle schema modifications, and other administrative operations. Hence, we have created a ”DBASQL” dataset having around 2500 pairs of natural language and SQL queries covering DDL, DML, and DCL queries for DBA requirements.

DBASQL Dataset contains natural language questions and corresponding SQL queries addressing DBA tasks like schema creation and modifications, updation of table contents, managing user permissions, etc. as listed in Table 1. These queries are divided into three different categories: DBA-related

DDL, DML, and DCL queries. This dataset also includes natural language questions that do not explicitly mention SQL clause names, making it easier for users to understand and interact with. By avoiding direct references to SQL clauses, the DBA experience will be improved. As a result, this dataset helps create a more natural and effective interaction between the DBA and the database system.

The count of queries is sufficient to train any LLM model which is given in Table 2, covering all DDL, DML, and DCL queries.

The screenshot of the DBASQL dataset is shown in Figure 2.



```
  {
    "input_text": "Increase the size of the 'company_name' column to 100 characters
in the StudentsPlacement table",
    "target_text": "ALTER TABLE StudentsPlacement ALTER COLUMN company_name TYPE
VARCHAR(100);"
  },
  {
    "input_text": "Rename the column 'salary' to 'monthly_salary' in the Professors
table",
    "target_text": "ALTER TABLE Professors RENAME COLUMN salary TO monthly_salary;"
  },
  {
    "input_text": "Add a new column 'course_code' to the Courses table",
    "target_text": "ALTER TABLE Courses ADD course_code VARCHAR(10);"
  },
  {
    "input_text": "Drop the 'enrollment_date' column from the Enrollments table",
    "target_text": "ALTER TABLE Enrollments DROP COLUMN enrollment_date;"
  },
```

Figure 2: DBASQL Dataset.

4 IMPLEMENTATION DETAILS AND EXPERIMENTAL SETUP

The following is the experimental setup and the model parameters used for fine-tuning T5 large.

Customized the DBASQL dataset for DBA-related queries and DataLoader classes tailored for the NLP to SQL task, which includes specific padding and tokenization requirements that can impact model performance. Training includes inference functions to infer query types from SQL queries, which are specific to the NL-to-SQL task. Finetuning of the T5 large model uses the following architecture parameters.

- Learning Rate: 1e-4 (0.0001),
- Number of Epochs: 20,
- Optimizer: AdamW optimizer,
- Scheduler: Learning rate scheduler with linear warm-up and linear decay,
- Batch Size: 4,
- Loss Function: The Cross-Entropy loss function is computed based on the output of the T5 model during training,
- Early Stopping: Patience of 5 epochs is used for early stopping. If validation loss does not improve for 5 consecutive epochs, the training stops.

The performance of finetuned and modified T5 is measured based on the following two parameters.

Exact Match Accuracy (EMA): It compares the expected query with the predicted SQL query to check whether they match each other. It is concerned with the syntactical correctness of the generated SQL. The EMA is calculated using Equation 1 as follows:

$$ACCema = Nema/n \tag{1}$$

Logical Accuracy(LA): It checks whether the generated SQL retrieves the correct data semantically, even if the SQL structure differs. The LA is calculated using Equation 2 as follows:

$$ACCla = Nla/n \tag{2}$$

Where n is the number of examples.
Nema- Number of predicted queries that are **syntactically** similar to the expected SQL query.
Nla- Number of predicted queries that are **logically** similar to the expected SQL query.

5 RESULTS AND DISCUSSION

We have proposed the novel DBASQL dataset for Database administration queries with the intention of handling the full breadth of SQL queries effectively. Also, experimenting with the proposed T5 model on a variety of NL questions will justify the strength of the proposed model. The training and validation loss graph for the fine-tuned T5 model is presented in Figure 3. The comparison of exact match accuracy(EMA) and logical accuracy(LA) for DDL, DML, and DCL queries is shown in figures Figure 4, Figure 5, and Figure 6, respectively. The proposed model is observed to be better at predicting DBA queries even without using the schema of the underlying tables.

The logical accuracy for DDL queries like CREATE TABLE and RENAME TABLE is more than the exact match accuracy as the primary keys are assigned automatically by the model as shown in Figure 4. Also, the renaming of the tables is done dynamically by the model.

Table 1: Description of Queries (DBA Related).

Category	Query Type	Description
DDL	`CREATE`	Used to create database objects like tables, indexes, and views.
	`ALTER`	Modifies the structure of an existing database object. Common operations include:
		`ADD`: Adds a new column or primary key to an existing table.
		`DROP COLUMN`: Removes an existing column from a table.
		`RENAME COLUMN`: Renames a column in a table.
		`RENAME TABLE`: Renames a table.
		`MODIFY COLUMN`: Changes the data type or size of an existing column.
	`DROP`	Deletes an existing database object such as a table or an index.
	`TRUNCATE`	Removes all rows from a table but retains its structure.
DML	`SELECT`	Retrieves data from one or more tables.
	`INSERT`	Adds new rows of data to a table.
	`UPDATE`	Modifies existing data in a table.
	`DELETE FROM`	Removes rows from a table.
DCL	`GRANT`	Allowing users or roles to perform specific actions like SELECT, INSERT, DELETE, and UPDATE on database objects.
	`REVOKE`	Restricting users from accessing or modifying database objects after permissions are revoked.

Table 2: Count of Queries(DBA Related).

DDL Query Type	Count	DML Query Type	Count	DCL Query Type	Count
CREATE	250	INSERT	176	GRANT	94
ALTER MODIFY	63	UPDATE	164	REVOKE	80
ALTER RENAME TABLE	80	DELETE FROM	150		
ALTER RENAME COLUMN	86				
ALTER MODIFY	85				
ALTER DROP COLUMN	50				
ALTER ADD	105				
DROP	47				
TRUNCATE	60				
DESCRIBE	30				
OTHER	25				

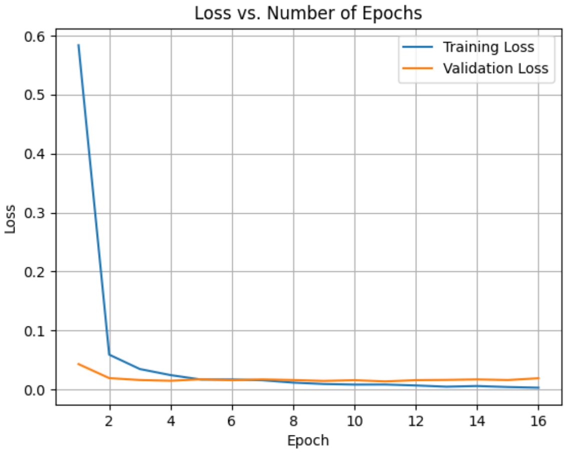


Figure 3: Loss Graph for Finetuned T5 Large.

The LA of DML queries like INSERT is more as alteration of the datatype and size are automatically done by the model as presented in Figure 5.

As shown in Figure 6, the EMA and LA are both the same for all natural language questions addressing DCL queries like granting and revoking user permissions like select, update, and delete on any table. Hence, the proposed model is observed to be better at predicting DBA queries even without using the schema of underlying tables.
The sample results addressing DDL, DML, and DCL queries are presented in Table 3, Table 4, and Table 5, respectively. For the majority of test data, the predicted query is the same as the expected query.

Also, for some test data, the model predicts the SQL clause without explicit mention in natural language questions, as shown in bold. As presented in Table 3, the proposed model creates the table without explicit mentions about the datatype and size of the table fields in the input natural language question. This makes the proposed model more robust. Also, for questions about creating the view, the model assigns the view names at run time.

Table 4 presents sample results for DML-related

Table 3: Sample Test Results on Finetuned T5 Large for DDL Queries.

Input Text	**Expected Output**	**Finetuned T5 Predicted Output**	**Remark**
Create a table named 'Emp' with columns: emp_id (INT), name (VARCHAR(100)), email (VARCHAR(255)), dept_id (INT), salary (DECIMAL(12, 2)).	CREATE TABLE Emp (emp_id INT, name VARCHAR(100), email VARCHAR(255), dept_id INT, salary DECIMAL(12, 2));	CREATE TABLE Emp (emp_id INT, name VARCHAR(100), email VARCHAR(255), dept_id INT, salary DECIMAL(12, 2));	Predicted is same as expected output.
Make table named 'Staffnew' with columns: staff_id, name, email.	CREATE TABLE Staffnew (staff_id INT, name VARCHAR(100), email VARCHAR(100));	**CREATE TABLE** Staffnew (staff_id INT PRIMARY KEY, name VARCHAR(255), email VARCHAR(255));	Model is generating SQL queries without their explicit mentions in NL question.
Add an index on the course_id column in the Course table.	CREATE INDEX idx_course_id ON Course(course_id);	CREATE INDEX idx_course_id ON Course(course_id);	Predicted is same as expected output.
Formulate a view that includes all professor's emails.	CREATE VIEW **ProfessorEmailList** AS SELECT email FROM Professors;	**CREATE** VIEW **ProfessorEmailListView** AS SELECT professor_id, email FROM Professors;	The view names are assigned dynamically.
Rename the column 'title' to 'title_paper' in the table 'ResearchPapers'	ALTER TABLE ResearchPapers RENAME COLUMN title TO title_paper;	ALTER TABLE ResearchPapers RENAME COLUMN title TO title_paper;	Predicted is same as expected output.
Change the name of the table 'ResearchPapers' to 'ResearchPapers1'	ALTER TABLE ResearchPapers RENAME TO ResearchPapers1;	**ALTER TABLE** ResearchPapers RENAME TO ResearchPapers1;	Model is generating SQL queries without their explicit mentions in NL question.
Modify the datatype of column 'enrollmentdate' to DATE in the table 'Enrollments'	ALTER TABLE Enrollments MODIFY COLUMN enrollmentdate DATE;	ALTER TABLE Enrollments MODIFY COLUMN enrollmentdate DATE;	Predicted is same as expected output.
Delete the column 'email' from the table 'Students'.	ALTER TABLE Students DROP COLUMN email;	ALTER TABLE Students **DROP COLUMN** email;	Model is generating SQL queries without their explicit mentions in NL question.
Add a new column 'graduation_year' to the Students table	ALTER TABLE Students ADD graduation_year INT;	ALTER TABLE Students ADD graduation_year INT;	Predicted is same as expected output.
Remove the Grades table, Drop the Grades table	DROP TABLE IF EXISTS Grades;	**DROP TABLE** IF EXISTS Grades;	Model is generating SQL queries without their explicit mentions in NL question
Delete all contents of the Students table	TRUNCATE TABLE Students;	**TRUNCATE TABLE** Students;	Model is generating SQL queries without their explicit mentions in NL question
Add a primary key constraint to the 'student_id' column in the 'Students' table.	ALTER TABLE Students ADD PRIMARY KEY (student_id);	ALTER TABLE Students ADD PRIMARY KEY (student_id);	Predicted is same as expected output.
Describe the Grades table	DESCRIBE Grades;	DESCRIBE Grades;	Predicted is the same as expected output.

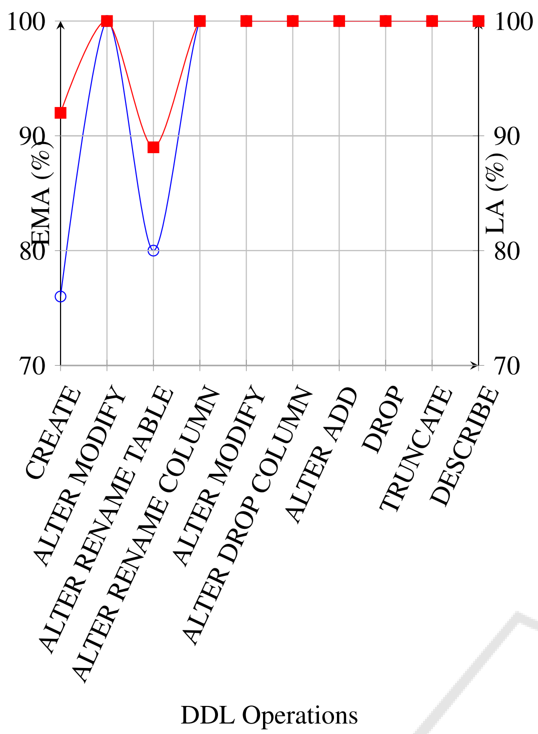


Figure 4: Comparison of EMA and LA for DDL queries.

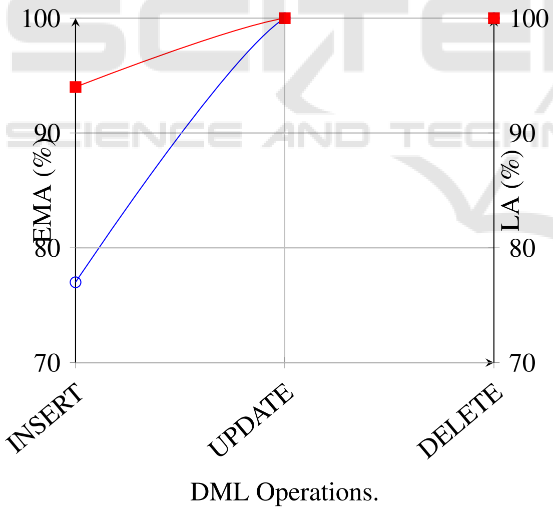


Figure 5: Comparison of EMA and LA for DML queries.

natural language questions. Here, the model predicts the correct SQL clauses without explicitly mentioning the SQL clause in the input natural language question, e.g. for inserting the values in the database table without an explicit mention of the ”INSERT” word in a natural language question, the model is still predicting the clause precisely.

Sample results of DCL queries are presented in Table 5. As presented in Table 5 the model is granting and revoking user permissions accurately on database tables.

Hence, the proposed model was found to achieve an EMA **94.74%** and LA **97.2%** with early stopping epoch number 14 using the proposed fine-tuned T5 large model without using the table schema as input.

6 CONCLUSION AND FUTURE SCOPE

This research work aimed to translate DBA-related natural language questions into SQL queries. We have proposed and validated fine-tuned T5 on a diversified customized DBASQL dataset, and we could achieve the exact match accuracy **94.74%** and logical accuracy **97.2%** without using the table schema as input. The proposed model can be effectively used in varying natural language users as well as DBA needs.
The same is justified and validated by the contribution in the form of a new proposed DBASQL dataset covering all varieties of natural language questions without explicitly mentioning the SQL clause. This dataset can be easily combined with the available heterogeneous datasets to cover the full breadth of SQL operations effectively for performing natural language to SQL translations.
In the future, we can continue to improve the efficiency of LLMs to handle more complex, ambiguous, and multi-turn queries.

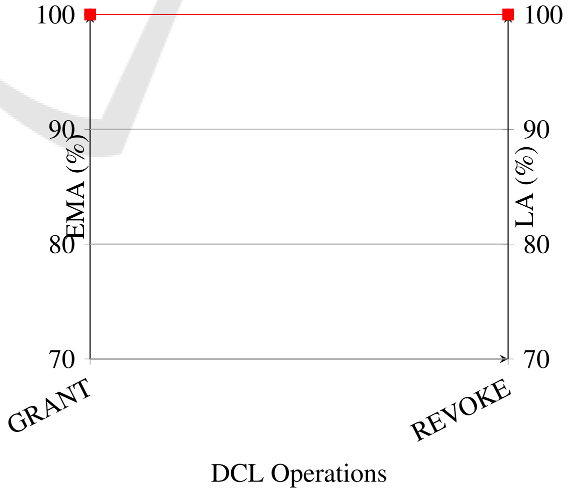


Figure 6: Comparison of EMA and LA for DCL queries.

Table 4: Sample Test Results on Finetuned T5 Large for DML Queries.

Input Text	Expected Output	Finetuned T5 Predicted Output	Remark
Delete a course having id 21	DELETE FROM Courses WHERE course_id = 21;	DELETE FROM Courses WHERE course_id = 21;	Predicted is same as expected output.
Change the title of a research paper to 'Advancements for Robotics', having paper id 30.	UPDATE ResearchPapers SET title = 'Advancements for Robotics' WHERE paper_id = 30;	**UPDATE** ResearchPapers SET title = 'Advancements for Robotics' WHERE paper_id = 30;	Model is generating SQL queries without their explicit mentions in NL question.
Insert a student with ID 11, name 'John', email 'john@example.com', department id 1, advisor id 101, gpa 3.5.	INSERT INTO Students (student_id, name, email, department_id, advisor_id, gpa) VALUES (11, 'John', 'john@example.com', 1, 101, 3.5);	INSERT INTO Students (student_id, name, email, department_id, advisor_id, gpa) VALUES (11, 'John', 'john@example.com', 1, 101, 3.5);	Predicted is the same as the expected output.
Add a student with ID 3, name 'Ronan', email 'ronan@example.com', advisor id 102, gpa 9.5, department id 4.	"INSERT INTO Students (student_id, name, email, department_id, advisor_id, gpa) VALUES (3, 'Ronan', 'ronan@example.com', 4, 102, 9.5);"	**INSERT INTO** Students (student_id, name, email, advisor_id, gpa, department_id) VALUES (3, 'Ronan', 'ronan@example.com', 102, 9.5, 4);	Model is generating SQL queries without their explicit mentions in NL question.

Table 5: Sample Test Results on Finetuned T5 Large for DCL Queries.

Input Text	Expected Output	Finetuned T5 Predicted Output	Remark
Enable ALL permissions on the 'inventory' table for role 'stock_manager'.	GRANT ALL PRIVILEGES ON inventory TO stock_manager;	**GRANT ALL PRIVILEGES** ON inventory TO stock_manager;	Predicted is the same as the expected output without explicit mentions in the NL question.
Allow user 'vidya' to SELECT data from the 'logs' table.	GRANT SELECT ON logs TO vidya;	**GRANT** SELECT ON logs TO vidya;	Predicted is same as expected output.
Take away UPDATE privileges on the 'inventory' table from role 'manager'.	REVOKE UPDATE ON inventory FROM manager;	**REVOKE** UPDATE ON inventory FROM manager;	Predicted is the same as the expected output without explicit mentions in the NL question.
Remove all access on the 'departments' table for the 'admin' role.	REVOKE ALL PRIVILEGES ON departments FROM admin;	**REVOKE ALL PRIVILEGES** ON departments FROM admin;	Predicted is the same as the expected output without explicit mentions in the NL question.

REFERENCES

A. Narechania, A. S. and Stasko, J. (2021). Nl4dv: A toolkit for generating analytic specifications for data visualization from natural language queries. *IEEE Transactions on Visualization and Computer Graphics*.

B. Wang, R. Shin, X. L. Y. P. and Richardson, M. (2020). Rat-sql: Relation-aware schema encoding and linking for text-to-sql parsers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7567–7578, Online.

C. Liu, Y. Han, R. J. and Yuan, X. (April 2021). Advisor: Automatic visualization answer for natural-language question on tabular data. In *14th IEEE Pacific Visualization Symposium, PacificVis*, page 11–20, Tianjin, China.

C. Raffel, N. Shazeer, A. R. K. L. S. N. M. M. Y. Z. W. L. and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*.

J. Devlin, M.-W. Chang, K. L. and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*.

K. Ahkouk, M. M. and Ennaji, M. (2021). Data agnostic roberta-based natural language to sql query generation. In *IEEE 6th International Conference for Convergence in Technology (I2CT)*.

K. Z. Hu, S. N. S. G. and et al. (2019). Viznet: Towards a large-scale visualization learning and benchmarking repository. In *Proceedings of CHI Conference on Human Factors in Computing Systems*.

M. A. Khan, M. S. H. Mukta, K. F. N. M. F. S. S. M. M. J. M. J. A. M. E. A. and Azam, S. (2024). A review on large language models: Architectures, applications, taxonomies, open issues and challenges. *IEEE Access*.

N. T. K. Le, N. Hadiprodjo, H. E.-A. A. K. and Teshebaev, A. (2023). Recent large language models in nlp. In *22nd International Symposium on Communications and Information Technologies (ISCIT)*, Sydney, Australia.

Q. Li, L. Li, Q. L. and Zhong, J. (2020). A comprehensive exploration on spider with fuzzy decision text-to-sql model. *IEEE Transactions on Industrial Informatics*, 16(4):April.

T. Yu, R. Zhang, K. Y. D. S. W. I. Z. and Su, M. Y. (2019a). Cosql: A conversational text-to-sql challenge towards cross-domain natural language interfaces to databases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1962–1979, Hong Kong, China.

T. Yu, R. Zhang, K. Y. M. Y. D. W. Z. L. J. M. I. L. Q. Y. S. R. Z. Z. D. R. (2018a). Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In *Proc. Conf. Empirical Methods Natural Lang. Process.*, page 3911–3921.

T. Yu, M. Yasunaga, K. Y. R. Z. D. W. Z. L. D. R. (2018b). Syntaxsqlnet: Syntax tree networks for complex and cross-domain text-to-sql task. In *Proc. Conf. Empirical Methods Natural Lang. Proces.*, page 1653–1663.

T. Yu, Z. Lin, L. T. D. S. W. I. Z. and Su, M. Y. (2019b). Sparc: Cross-domain semantic parsing in context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4511–4523, Florence, Italy.

T. Yu, Z. Li, Z. Z. R. Z. and Radev, D. (2018). Type-sql: Knowledge-based type-aware neural text-to-sql generation. In *16th Annual Conference of the North American Chapter of the Association for Computational Linguistics*, New Orleans.

Trummer, I. (2022). Codexdb: Generating code for processing sql queries using gpt-3 codex. In *Proceedings of the International Conference on Artificial Intelligence*.

V. Zhong, C. X. and Socher, R. (2017). Seq2sql: Generating structured queries from natural language using reinforcement learning. *arXiv:1709.00103*.

Y. Li, Z. Su, Y. L. H. Z. S. W. W. W. and Zhang, Y. (2023). T5-sr: A unified seq-to-seq decoding strategy for semantic parsing. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Y. Luo, N. Tang, G. L. C. C. W. L. and Qin, X. (2021). Synthesizing natural language to visualization (nl2vis) benchmarks from nl2sql benchmark. In *International Conference on Management of Data*, page June 20–25. ACM.

Y. Luo, N. Tang, G. L. J. T. C. C. and Qin, X. (2022). Natural language to visualization by neural machine translation. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):January.

Z. Zhao, Y. Yang, X. H. and Liang, J. (2021). Smbop: Semi-autoregressive bottom-up semantic parsing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5903–5914, Online.