Intelligent Problem Solver in Database Systems based on Ontology Integration through Text-to-SQL

Duc Truong [1,2], Hung Nguyen [3,*], Nha P. Tran [4], Sang Vu [1,2], Hien D. Nguyen [5,2]

[1] Faculty of Information System, University of Information Technology, Ho Chi Minh city, Vietnam;,
[2] Vietnam National University, Ho Chi Minh city, Vietnam
[3] Faculty of Information Technology, Ho Chi Minh University of Education, Ho Chi Minh City, Vietnam;
[4] Campus in Ho Chi Minh City, University of Transport and Communications, Vietnam;
[5] Faculty of Computer Science, University of Information Technology, Ho Chi Minh city, Vietnam;
Emails: 21521971@gm.uit.edu.vn; hungnv@hcmue.edu.vn; hungv@hcmue.edu.vn; nhatp_ph@utc.edu.vn; sangmv@uit.edu.vn; hiennd@uit.edu.vn
* Correspondence: hungnv@hcmue.edu.vn

Abstract

The knowledge of courses can be represented by using ontology to create intelligent educational systems. This study proposes the Onto-Linking model as a knowledge framework that expresses the knowledge of the inputted schema to investigate the schema linking problem of the Text-to-SQL model. It combines the ontology with the structure of the schema. The proposed ontology is utilized to encapsulate the semantics of the intellectual elements of the schema, such as the table names, column names, foreign/primary key restrictions, and information about the probing schema connection. Therefore, the model makes it easier to accurately translate natural language questions into SQL queries. It improves query creation, helps with error handling, and supports query validation by helping the model better grasp the query's intent. The outcomes of the pedagogically oriented model aimed at guiding learners to comprehend the process of reasoning to attain the respective solution.

Keywords: Ontology; T5; Text-to-SQL; Schema linking; E-learning.

1. Introduction

In the Industrial Revolution 4.0, e-learning is a useful method for online learning. E-learning is an online educational system that uses technological, electronic devices via the Internet [1, 2]. Using the e-learning education system, students can reference materials and exchange with faculty to acquire knowledge effectively. Ontology is a useful tool for representing the knowledge of courses to build intelligent systems in education [3], such as query systems through semantic searching, intelligent problem solvers for tutoring solving methods of exercises [4], and the evaluation system for intellectual level of learners [5, 6]. It serves as the underlying structure that facilitates effective organization, representation, and retrieval of information, thereby enhancing the design and delivery of e-learning experiences.

The ontology-data model is an ontology-integrating database for representing the intellectual of courses [7]. This model is used to solve knowledge-based query problems such as: searching articles/chapters, searching information based on knowledge classification, and recommending relevant knowledge. The Fundamental of Database Systems course is an important subject in the Information Technology curriculum at the university level [8]. Within the scope of this course, learning about the concept and application of SQL (Structured Query Language) is important when discussing database-related topics. Because SQL can be used to solve issues, manipulate data, and extract valuable information, it is a crucial topic that can aid students in understanding database management concepts. SQL is frequently challenging to teach and learn due to the abstraction of conditions, complicated queries, and poorly defined errors.

To good effectiveness in supporting the learning of SQL in the Fundamental of Database Systems course, the intelligent SQL problem solver system that supports the solving of tasks must meet the following criteria:

Doi: https://doi.org/10.54216/FPA.150211
Received: August 21, 2023 Revised: December 17, 2023 Accepted: April 11, 2024

- Query generation: A text-to-SQL model can accept a natural language query as input and output the equivalent SQL query. This enables students to express their queries in their natural language and the model helps translate them into the proper SQL syntax. Bridging the gap between natural language and SQL, makes learning easier.
- Error correction: When students try to build SQL queries, a text-to-SQL model can offer immediate feedback by pointing out syntax mistakes or making recommendations to improve. This facilitates a better understanding of SQL syntax and query building by learners in comprehending and fixing their errors.

The text-to-SQL model is designed to address issues with the translation of inquiries from natural language into structured SQL languages [9, 10]. This model has gained popularity recently since it enables effective database information extraction without the need for technical knowledge.

In this paper, a method for integrating the ontology and the structure of schema called the Onto-Linking model, is proposed as a knowledge framework that represents the knowledge of the input schema. The proposed ontology is used to represent the semantics of intellectual components in the schema, including foreign/primary key constraints, table names, and column names/datatypes. This probing schema linking information is integrated into a knowledge graph, which constitutes a foundational resource for the subsequent fine-tuning of the T5 model. The proposed knowledge graph facilitates the translation of natural language queries into SQL queries accurately. It enhances the model's understanding of the query intent, improves query generation, and aids in error handling and query validation.

From that schema linking information, the information of the schema into the embedding system of the model ensures that entity references in natural language queries are suitably aligned with the specified schema columns or tables. With the trained Text-to-SQL model, we construct an intelligent SQL problem solver system to assist students in creating SQL queries from their respective schema and the natural language question. Besides, we incorporate error-correction technology that can detect users' SQL query problems based on their input criteria and provide corrections for them.

2. Related work

Text-to-SQL systems bridge the gap between natural language queries and SQL queries, facilitating efficient interaction with databases [9, 11]. Application of sequence-to-sequence (seq2seq) models is a well-known strategy in Text-to-SQL modeling. Those models developed for machine translation have been effectively applied to the Text-to-SQL challenge, displaying their adaptability and efficiency [11]. Schema generalization is difficult for three related reasons. Any text-to-SQL parsing model must first encrypt the schema into representations that can be used to decode a SQL query that might use the specified columns or tables [11, 12]. Besides, the information about the schema, including the types of columns, foreign key relationships, and primary keys used for database joins, should be encoded in these representations. In the studies of solving this issue, schema linking has been learned to align entity references in the natural language query to the intended schema columns or tables [13]. It is essential to achieve domain generalization.

The Transformer model, which is the best state-of-the-art seq2seq model until now, was developed in [14, 15]. It uses a self-attention mechanism that allows it to recognize long-range dependencies and contextual relationships among words in a sequence. The Transformer model efficiently handles both short- and long-range dependencies by using self-attention layers rather than recurrent neural networks (RNNs) [16], allowing parallel processing and speeding up training. The Transformer model has since been used as the basis for many other models, and it has displayed outstanding performance in several language-related tasks.

T5 model is the building upon the Transformer model [17]. The text-to-text framework introduced in T5 model integrates different natural language processing tasks under a common paradigm. It makes use of a single Transformer model that has been trained in a multitask learning scenario rather than creating task-specific architectures. The model achieves outstanding performance on a variety of benchmarks for natural language processing after being pretrained on a big corpus of different data and tailored for downstream applications [18].

The availability of high-quality datasets is crucial for the development and evaluation of Text-to-SQL models. Researchers have contributed to the field by creating annotated datasets dedicated to Text-to-SQL model training and benchmarking. Notable datasets like WikiSQL, derived from Wikipedia tables, and Spider, sourced from real-world web applications, have advanced the state-of-the-art in Text-to-SQL modeling and enabled fair comparisons of methodologies. WikiSQL offers a diverse collection of natural language inquiries and corresponding SQL queries, covering fundamental SQL operations [19]. Spider, on the other hand, provides more complex queries with JOINs, subqueries, and multi-step operations, along with various database schemas, offering a realistic and

Doi: https://doi.org/10.54216/FPA.150211
Received: August 21, 2023 Revised: December 17, 2023 Accepted: April 11, 2024

challenging training environment [20, 21]. Additionally, the SYN dataset (the dataset of synonym) has been used to enhance training by synthesizing data and generating SQL queries from natural language templates [22]. However, current parsers face challenges in accurately linking entity mentions to schema elements, particularly when dealing with different surface forms of mentions and columns/tables. The need for robust schema-linking relations remains, as cutting-edge parsers struggle with synonym substitution and real-world scenarios.

In [23], the authors present a method for developing an intelligent querying system for e-learning, in which its structure is built on the combination of an ontology and the two-layer knowledge graph. The model has been proven that it could graphically display the links between distinct knowledge domain components and enhance adaptability. Besides, the model was utilized to construct an intelligent querying system for the Fundamentals of Database Systems course in the IT curriculum [7, 24]. The efficiency that this model delivers, prompted us to integrate the knowledge graph into the Text-to-SQL parse.

Given the challenges, advantages, and resources outlined above, we have decided to integrate an ontology-based approach to extract the distinctive relationships within the database schema. Our approach utilizes the semantic vectors derived from this approach to fine-tune the T5 model, addressing the Text-to-SQL problem with a breakthrough in understanding the semantic correlation between queries and the database schema.

3. Ontology-based knowledge model for Text-to-SQL problem

In this work, a knowledge model based on ontology for translating text to SQL language is called the Onto-linking model. This model is constructed on the Spider and SYN datasets, from the T5-large checkpoint of T5 model [17]. When a user enters input, the database schema is integrated with the question as the input rather than just encoding the questions themselves since we wanted the model to produce the SQL query for a specific database. Then, the Onto-Linking model parses the relevant input into a keyphrase graph that comprises information about the input database schema, including the columns, tables, constraints, and their relationships with the input tokens. After that, the transformer-based encoder-decoder architecture is utilized as the core of the proposed knowledge model. To embed the following keyphrase graph, the same technique is used as relation-aware self-attention [18], a model for embedding semi-structured input sequences in a way that simultaneously encodes both pre-existing relational structure in the input and induced known relations between sequence elements in the same embedding [25].

3.1 Integrating ontology Onto-Linking model and the keyphrase graph

A. Structure of Onto-Linking model

Onto-Linking model is a model for representing the combining knowledge of natural language queries, components of database schema attached with its metadata details. This model is built based on the structure of Rela-model [26], which is an effective model for representing the knowledge domain about relations. Rela-model combines cognitive components with knowledge-related inference processes, including concepts, relationships, and operators between ideas; it has been used to create intelligent problem solver about solid geometry in high school mathematics for students. Although being constructed based on the Rela-model, the Onto-linking model has undergone substantial architectural improvements by incorporating a two-layer knowledge graph to depict semantic relationships more effectively among its components. By leverages ontologies and semantic information to make linkages between textual queries and the underlying database structure, it utilizes the ontological relationships and constraints encoded in the keyphrase graph to infer and disambiguate the mappings between textual terms and schema element, therefore offers a more robust and accurate understanding of natural language queries and facilitates the production of accurate SQL representations for successful interaction with the database.

Definition 3.1: The improvement structure of Rela-model for representing the combining knowledge of natural language queries, components of database schema is a tube that includes three components:

(C, R, Rules)

In which, **C**-set is a set of input components includes the natural language query and the database schema information as Data Definition Language (DDL). The **R**-set is a collection of relations between question tokens, schema components and their details. The rules in the **Rules**-set serve as representations for assertions, theorems, guiding principles, formulas, and other notions to extract keyphrases and measure the extent of semantic similarity between components. The structure of **(C, R, Rules)** has been described in [23, 26].

To represent more the semantic relations between the input elements, a dictionary of keyphrases, which are extracted from the natural language query using the approach mentioned in [27], combines with the database schema components that are parsed from the Data Definition Language (DDL) statement. They are integrated to the architecture of Onto-Linking model, which is defined in Def. 3.2.

Doi: https://doi.org/10.54216/FPA.150211
Received: August 21, 2023 Revised: December 17, 2023 Accepted: April 11, 2024

Definition 3.2: Onto-Linking model is a combination between the improvement of Rela-model and the dictionary of keyphrases representing semantic relations. Onto-Linking model is a tube:

$$\textbf{(C, R, Rules)} \oplus \textbf{(Q, <T, C>)}$$

In which, **(C, R, Rules)** is an improvement model of Rela-model as Definition 3.1.

(Q, <T, C>) is the structure of dictionary of keyphrases representing semantic relations. It is an essential component of the integrated ontology-based Onto-Linking model, designed to bridge the gap between natural language questions and structured database queries.

- **Q**-set is a sequence of tokens extracted from natural language questions.
- **<T, C>** is the database schema which consists of tables **T**-set and columns **C**-set. **T**-set and **C**-set are the sets of table names and column names present in the database schema, respectively, whereas each table name t_i and column name c_i contains multiple words that were extracted by the tokenizer.

B. Keyphrase graph for Onto-Linking model

The keyphrase graph based on the keyphrase dictionary helps to enrich the semantic relations between the keyphrases that are extracted from the last section. A keyphrase graph is a structured representation of text that captures relationships between entities. It organizes information in the form of nodes and edge to create a semantic network [28], therefore offers a semantic context that assists in producing precise and contextually appropriate SQL queries. The keyphrase graph for the Onto-Linking model is a two-layer graph:

- The first layer represents the entities present in the inputted natural language text, such as tokenized words and syllables. Each entity is represented as a node in the graph, the edges encapsulate the semantic sharing between them.
- The second layer organizes the schema details. It displays the structure and relationships of the database schema. The nodes in this layer correspond to tables, columns, and datatypes, while the edges stand in for the connections between each table-column, table-table and column-column pairs.

Definition 3.3: Given the knowledge domain K = **(Q, <T, C>)** as the structure of the keyphrase dictionary. Using the same knowledge graph base structure of the Rela-KG model in [23], the keyphrase graph is designed as a tube:

$$\textbf{KG: = (V, E)}$$

The vertices in **V** correspond to subword tokens, which are essential linguistic units derived from the keyphrase dictionary, capturing fine-grained information, and aiding in comprehensive language understanding. These subword tokens serve as the keyphrases of the query and schema, encapsulating significant concepts and terms for subsequent processing. Each vertex in **V** can be visualized as tube consisting of two essential components (*Attributes, Type*). Where, the *Attributes* set represents the important descriptive and metadata related to the vertex such as: an embedded semantic vector of the keyphrase, a datatype corresponding to the column. Besides, the *Type* set represents the truth type of the vertex including *Question*, *Column*, *Table.*

The edges in **E** establish connections between the keyphrases, manifesting semantic associations, syntactic dependencies, or co-occurrence patterns between them. There are two kinds of edges in **E:**

$$\mathbf{E = E_{attrs} \cup E_{objs}}$$

$\mathbf{E_{attrs}}$ represents the edges connecting keyphrases to their corresponding attributes, capturing essential characteristics or properties associated with the keyphrases. These attributes can include syntactic information, such as part-of-speech tags or grammatical relationships, as well as semantic attributes that convey the meaning and context of the keyphrases. On the other hand, $\mathbf{E_{objs}}$ represents the edges connecting keyphrases to other keyphrases, reflecting the semantic relationships and associations between them. These relationships can encompass similarities, co-occurrence patterns, or hierarchical structures that establish a more comprehensive representation of the underlying information.

Each edge in **E** can be thought of as a tube made up of the four basic parts (*Attributes, Properties, Start, End*), in which, *Attributes* is a set include crucial descriptive data and metadata pertaining to the edge, offering insightful context and specifics on the semantic link between the keyphrases it connects. The dynamics and functionality of the linked knowledge are influenced by the *Properties* set, which determine certain features and behavior of the edge within the graph. The *Start* and End components specify the source and target keyphrases connected by the edge, indicating the direction of information flow and allowing a thorough depiction of the interconnectivity of the textual data.

By incorporating both $\mathbf{E_{attrs}}$ and $\mathbf{E_{objs}}$, the keyphrase graph can leverage a rich network of interconnected keyphrases and their attributes. This comprehensive approach to constructing the edges in the keyphrase graph enhances the system's ability to capture and interpret the intricate relationships embedded within the database schema information and the natural language query. Figure 1 shows a sample keyphrase graph extracted from the user input.

Doi: https://doi.org/10.54216/FPA.150211
Received: August 21, 2023 Revised: December 17, 2023 Accepted: April 11, 2024

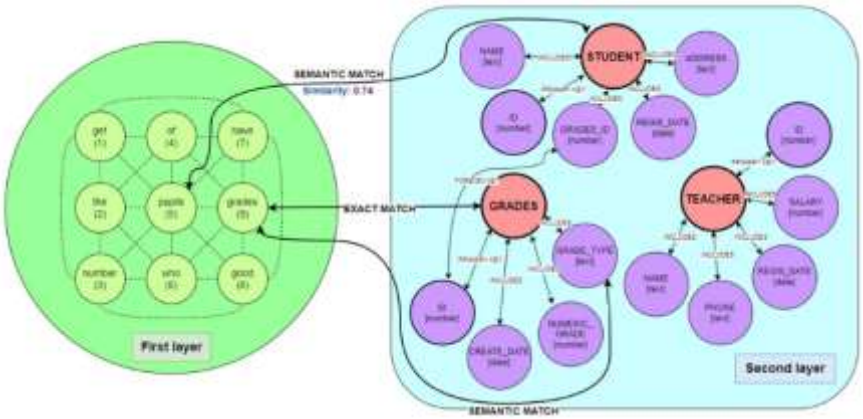


Figure 1: The structure of keyphrase graph

3.2 Training model for Text-to-SQL

A. Datasets

To develop and assess Text-to-SQL models, high-quality dataset availability is essential. Our research involves the rigorous training and comprehensive assessment of these models using three benchmark datasets.

WikiSQL dataset is one popular set of data that is used to train Text-to-SQL models - which is derived from a corpus of Wikipedia tables, offers a sizable collection of natural language inquiries along with corresponding SQL queries [19]. The dataset includes a broad range of fundamental SQL operations, such as SELECT, WHERE, and aggregation, and has been manually annotated with precise SQL queries.

Spider is another notation dataset, derived from real-world web applications. It offers a special selection of hard Text-to-SQL models, elaborate SQL queries, and sophisticated features like JOINs, subqueries, and multi-step operations [20]. The dataset includes multiple database schemas from different fields, offering a realistic and demanding training environment for Text-to-SQL models. **Spider** has acquired appeal as a benchmark dataset due to its portrayal of real-world complexity and consistent annotations [21].

SYN dataset has been used to improve Text-to-SQL model training [22]. The **SYN** dataset focuses on synthesizing data to provide extra training instances. By automatically producing SQL queries from natural language templates and matching database schemas, the **SYN** dataset augments the training data available for Text-to-SQL models. This synthetic technique broadens the range of query types and variants, thus enhancing the model's capacity to handle various SQL queries.

B. Training model

The training model for Text-to-SQL is fine-tuned from the t5-large checkpoint of the T5 model [17, 18]. The T5 model has derived its architecture from the Transformer model, which has had a profound impact on the field of natural language processing by utilizing self-attention mechanisms to capture long-range dependencies in sequential data more efficiently [29]. To embed the semantic relationships extracted from the keyphrase graph mentioned above, we apply the method proposed in [18], which is proven effective for encoding position and segment information into the Transformer models.

Doi: https://doi.org/10.54216/FPA.150211
Received: August 21, 2023 Revised: December 17, 2023 Accepted: April 11, 2024

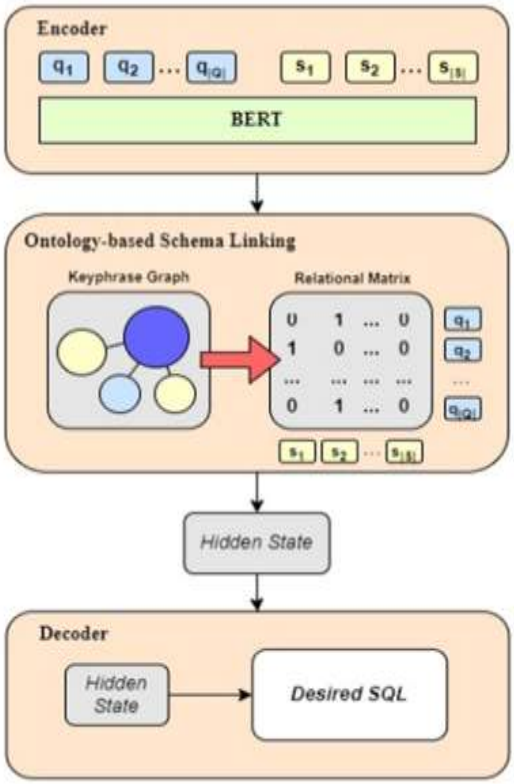


Figure 2: The architecture of training model for Text-to-SQL based on T5 model.

In Figure 2, the same fine-tuning strategy is followed as the original T5. AdamW has been adapted as the optimizer to update the model's parameters. This optimizer prevents overly penalizing large weights by decoupling weight decay from the optimization step and applying it directly to the model's parameters [30]. It calculates the gradients of the loss function with respect to the model parameters and modifies the first and second moment estimations throughout each training step. Then, it applies the bias-corrected estimates to update parameters, while separately subtracting the weight decay term from parameters.

4. Problems and Algorithms

4.1 Generating SQLs on the integrated ontology-based model

The Intelligent Problem Solver in Database Systems is designed to address query generation problems which offer students a positive learning environment and helpful support for mastering SQL. When a user inputs a natural language query and define the corresponding database schema with the DDL statement. At first, the system verifies the input DDL statement and extracts the schema information including table and column metadata. Following that, the system extracts and examines the natural slanguage query's keyphrases along with the database schema information extracted from the first step, using the subword tokenizer method proposed in [31]. It will then store all the tokens by the structure of keyphrase dictionary and emphasizes the hierarchy relationships between keyphrases using the keyphrase graph model proposed in section 3.2. Table 1 describes the types of edges the keyphrase graph as Section 3.1.B. Those types are also types of relations in the knowledge domain.

Table 1: Types of relations

Order	Relation	Attributes	Description
1	FOREIGN-KEY(v_1, v_2)	v_1, v_2 ∈ **C**	v_1 is the foreign key for v_2
2	FOREIGN-KEY(v, t)	v ∈ **C**, t ∈ **T**	v is the foreign key that references to a column in t
3	PRIMARY-KEY(v_1, v_2)	v ∈ **C**, t ∈ **T**	v is the primary key of t
4	CONTAINS(t, v)	v ∈ **C**, t ∈ **T**	t is the column of v
5	MATCH(q, s)	q ∈ **Q**, s ∈ **T** × **C**	the query component q has semantic similarity with the schema component s.
6	MATCH(s, q)	q ∈ **Q**, s ∈ **T** × **C**	the schema component s has semantic similarity with the query component q.

Doi: https://doi.org/10.54216/FPA.150211
Received: August 21, 2023 Revised: December 17, 2023 Accepted: April 11, 2024

To determine a query token $q \in \mathbf{Q}$ has semantic similarity with a schema component $s \in \mathbf{T} \times \mathbf{C}$, both s and q are fed into the pretrained T5 model checkpoint to retrieve their contextualized representation, which are denotes as $\mathbf{h_s}$ and $\mathbf{h_q}$ respectively. After that, we calculate the correlation between the schema item **s** and the question token **q** by quantifying the distance between $\mathbf{h_s}$ and $\mathbf{h_q}$ using the Poincaré ball function:

$$d(s,q) = \operatorname{arcosh}\left(1 + \frac{2.\|s-q\|^2}{\left(1-\|s\|^2\right)\left(1-\|q\|^2\right)}\right) \quad (1)$$

where, ||.|| is Euclidean, and *arcrosh* is the inverse hyperbolic cosine function. By employing the Poincaré ball function, semantic similarity computation in hyperbolic space allows for more effective modeling of hierarchical structures and capturing fine-grained semantic relationships.

After computing all the similarity scores between items with Poincaré Ball function, we utilize *Min-Max scaler* for making it easier to compare different features with each other.

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (2)$$

4.2 Algorithm for determining the relations between query and database schema

Given a knowledge domain K = **(C, R, Rules)** $\oplus$ **(Q, <T, C>)** as Onto-Linking model, the natural language query *q*, and the DDL statement *s* that the define the corresponding database schema. This algorithm determines relations between query *q* and database schema *s*.

Input: The knowledge domain K as Onto-Linking model, query *q*, and DDL statement *s*

Output: A two-dimension relational matrix that represent the relationships between each component in **Q**-set and **S**-set

Algorithm 4.1

RelaMat := [[]] // A 2D matrix to represent the relationships between **Q**-set and **S**-set tokens
Schema-Tok := {} // Set of keyphrases of the schema information
Query-Tok := {} // Set of keyphrases of the natural language query
Schema-Vect := {} // Set of contextualize vectors of the schema tokens
Query-Vect := {} // Set of contextualize vectors of the query tokens

Step 1:

- Using the SentencePiece tokenizer mentioned in [34] to extract the subword tokens from *q*, then append the extracted tokens into *Query-Tok* set.
- Parse *s* using the method proposed in [29, 32] to extract its metadata information, then tokenizes them using the above technique and update the *Schema-Tok* set.
- Construct relations between keyphrases in *Schema-Tok* through relations in K.**R**, then update *RelaMat*

for *base* in *Schema-Tok* **do**
  for *targ* in *Shema-Tok* **do**
    if *base* <> *targ* **do**
      $RelaMat_{base,\ targ}$:= K.R$_{base}$[*targ*]

Step 2: Using the pretrained T5 model to embed the tokens in both *Schema-Tok* and *Query-Tok* sets into the semantic representation as vectors.

- *Schema-Vect* := Embedding(*Schema-Tok*)
- *Query-Vect* := Embedding(*Query-Tok*)
- **Update** *Schema-Vect*, *Query-Vect*

Step 3:

- Implement the Poincaré Ball function to compute the semantic similarity between tokens in *Schema-Tok* and *Query-Tok* in hyperbolic space.

for qtok in *Query-Tok* **do**
  for stok in *Schema-Tok* **do**
    $RelaMat_{qtok,stok}$:= d(*stok*, *qtok*)

- **Update** *RelaMat*

Step 4:

- Using *MinMaxScaler* method to scale and normalize matrix values within the range 0 and 1.
- Determine if there is a relation between a token in *Query-Tok* and a token in *Schema-Tok* using a threshold *t* that we determined out throughout testing:

$$RelaMat_{n,m} = \{1, if\ RelaMat_{n,m} > t\ 0, if\ RelaMat_{n,m} \le t \quad (3)$$

- **Update** *RelaMat*

Step 5: Return *RelaMat*

5. Testing and Experimental results

5.1 Testing

The designed Intelligent Problem Solver (IPS) in Foundation of Database Systems course, called *DxB Text to SQL* system, aims to evaluate its performance in addressing two fundamental challenges, including query generation and error correction. This system must satisfy the criteria of an Intelligent Problem Solver in education [33]. By harnessing the capabilities of the T5 language model and fine-tuning strategies, this system facilitates intuitive and efficient interactions between users and databases. Figure 3 gives a sample database schema.

Doi: https://doi.org/10.54216/FPA.150211
Received: August 21, 2023 Revised: December 17, 2023 Accepted: April 11, 2024

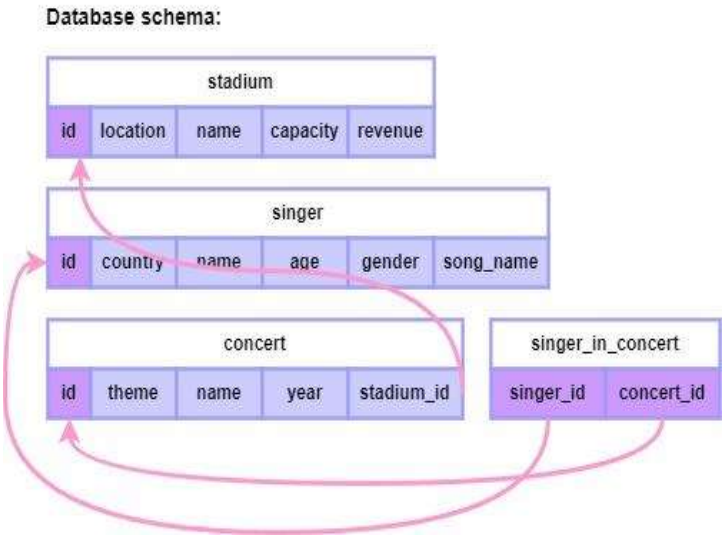


Figure 3: A sample database schema

To illustrate the simple interface of the system and provide a practical demonstration of its usage, we have mapped the database schema above to its corresponding DDL commands and developed some challenging natural language questions for inputting into the system to facilitate experimentation. The generation results are represented in Table 2.

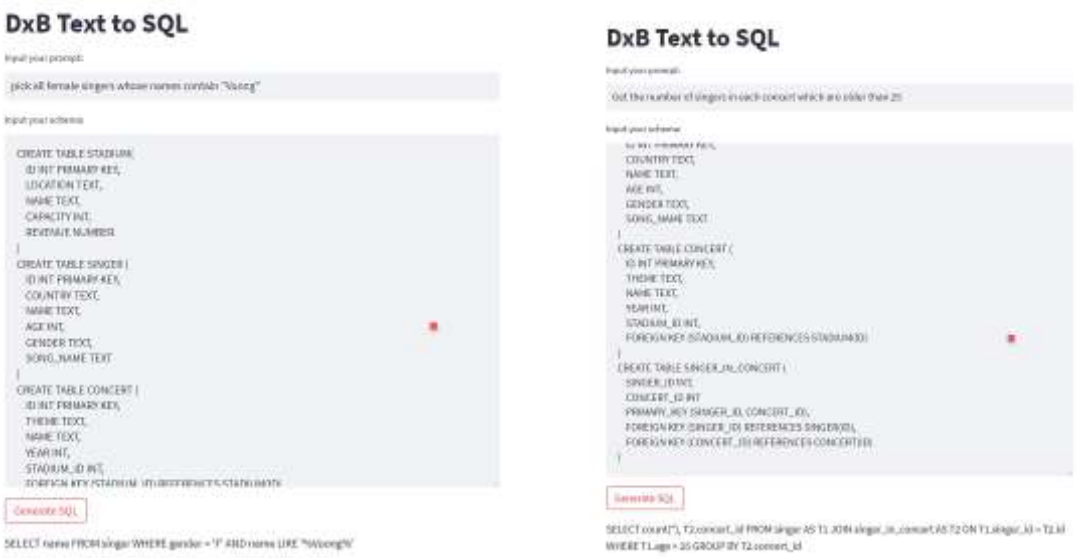
Table 2: Generated results in SQL language

Order	Natural Language Question	Generated SQL
1	Pick all female singers whose names contain "Vuong"	SELECT name FROM singer WHERE gender = 'F' AND name LIKE '%Vuong%'
2	List all concerts that are hosted in a stadium with more than 80 seats	SELECT T2.name FROM stadium AS T1 JOIN concert AS T2 ON T1.id = T2.stadium_id WHERE T1.capacity > 80
3	Compute the total revenue of each stadium	SELECT sum(revenue), id FROM stadium GROUP BY id
4	Get the number of singers in each concert who are older than 25	SELECT count(*), T2.concert_id FROM singer AS T1 JOIN singer_in_concert AS T2 ON T1.singer_id = T2.id WHERE T1.age > 25 GROUP BY T2.concert_id

Table 2 presents four challenging cases categorized based on their increasing complexity levels. The initial case involves a combination of a WHERE clause and an operator, requiring the system to handle basic conditional statements. In the second case, the query necessitates the integration of a conditional clause with a JOIN clause, posing a higher level of complexity. The third case demands the generation of a query featuring an aggregate clause, adding a new layer of intricacy to the system's task. Lastly, the fourth case represents the most demanding scenario, wherein the system must generate a complex query combining an aggregate clause, a conditional clause, and a JOIN clause, showcasing the model's ability to tackle advanced query structures.

Figure 4 are results of Case 1 and Case 4 which were SQL sentences generated by the designed program.

Doi: https://doi.org/10.54216/FPA.150211
Received: August 21, 2023 Revised: December 17, 2023 Accepted: April 11, 2024



a) Result of Case 1 b) Result of Case 4

Figure 4: The results were generated by the designed program.

5.2 Experimental results

The Intelligent Problem Solver in Foundation of Database Systems course has been tested in 04 kinds of exercises which were collected from [8]:

- Kind 1: This category comprises basic SELECT queries, which extract queries involving simple data retrieval from a single table.
- Kind 2: This kind encompasses JOIN queries, which identify queries that require combining information from multiple tables using different JOIN operations.
- Kind 3: This category gathers aggregate queries, focusing on queries that involved computing summary statistics using functions like SUM, COUNT, or AVG.
- Kind 4: This kind includes complex queries that combined elements from the previous three types, incorporating both JOIN and aggregate operations.

This systematic categorization enabled us to evaluate the system performance across various query complexities and gain valuable insights into its ability to handle a diverse range of SQL tasks, thereby contributing to the advancement and optimization of the system for practical applications. The results of this experiment are shown in Table 3.

Table 3: Result of experiments

Kind	Number of testing	Correctly	Rate
1	167	121	72.5%
2	150	98	65.3%
3	159	108	67.9%
4	103	46	44.7%
Total	**579**	**373**	**64.4%**

The obtained test results encompass a total of 579 questions. For the relatively straightforward problems of Kind 1, the system exhibits a commendable accuracy of 72.5%. However, as the complexity of the SQL statements increases in problems of Kind 2 and beyond, the accuracy of the system gradually declines. Particularly, in the most challenging problems in the fourth class (Kind 4), which involve a combination of the three afore mentioned kinds, the system achieves the lowest accuracy, reaching 44.7%. These findings underscore the system's limitations in effectively handling complex queries that necessitate intricate combinations of relational operations, JOIN clauses, and aggregate functions on new database schemas. Thus, highlights the need for further development and refinement to bolster the system's capacity to generate precise and optimal SQL statements for intricate tasks on open-domain real-world databases.

6. Conclusion

In this paper, an ontology-based technique to investigate the schema linking problem of Text-to-SQL model, which is called Onto-Linking model, is proposed. This model is a combination of ontology and keyphrase graph. Then, it was applied to gain a deeper understanding of the domain-specific database schema, thus enhancing the performance of the Text-to-SQL parser fine-tuned from the T5 language model. The proposed method is applied to represent the knowledge of Database Systems course in an IT curriculum at university level. Besides, an IPS in Foundation of Database Systems course is also designed, called DxB Text to SQL. This system can solve common

Doi: https://doi.org/10.54216/FPA.150211
Received: August 21, 2023 Revised: December 17, 2023 Accepted: April 11, 2024

kinds of exercises in the course's curriculum at the university level. In the future, an intelligent problem solver in database systems will be researched and developed to address the problems about query optimization, inefficient subquery handling, and limited knowledge of the database structure [35]. Moreover, they integrate the Intelligent Problem Solver in Foundation of Database Systems course into an e-learning platform to provide a dynamic and engaging learning environment that demystifies the complexity of SQL and database queries [36]. The desired platform may offer SQL tutorials, where users may adhere to detailed guidelines to comprehend SQL ideas, syntax and best practices. Thus, they will improve the learning process of querying on Database Systems. Additionally, the platform can incorporate practical exercises and challenges that allow learners to apply their knowledge in real-word scenarios [23, 37]. Therefore, the Intelligent Problem Solver in Foundations of Database Systems will be an online tutoring platform that is more sophisticated and engaging in promoting students learning.

Funding: "This research is funded by Ho Chi Minh City University of Education Foundation for Science and Technology under grant number CS.2023.19.19"

Conflicts of Interest: "The authors declare no conflict of interest."

References

[1] L. Wu, P. J. Hsieh, & S. M. Wu, Developing effective e-learning environments through e-learning use mediating technology affordance and constructivist learning aspects for performance impacts: Moderator of learner involvement. The Internet and Higher Education, 2022, vol. 55, 100871.
[2] A. N. Saleem, N. M. Noori, & F. Ozdamli, Gamification applications in E-learning: A literature review. Technology, Knowledge and Learning, 2022, vol. 27, no. 1, 139-159.
[3] H. Nguyen, N. Do, & V. Pham, A methodology for designing knowledge-based systems and applications. In Applications of Computational Intelligence in Multi-Disciplinary Research, 2022,159-185. Academic Press, Elsevier.
[4] H. D. Nguyen, D. A. Tran, H. P. Do, & V. T. Pham, Design an intelligent system to automatically tutor the method for solving problems. International journal of integrated engineering, 2020, vol. 12, no. 7, 211-223.
[5] I. A. Mastan, D. I. Sensuse, R. R. Suryono, & K. Kautsarina, Evaluation of distance learning system (e-learning): a systematic literature review. Jurnal Teknoinfo, 2022, vol. 16, no. 1, 132-137.
[6] T. T. Mai, H. D. Nguyen, T. T. Le, & V. T. Pham, An Intelligent Support System for the Knowledge evaluation in high-school mathematics by Multiple choices testing. In 2018 5th NAFOSTED Conference on Information and Computer Science (NICS) (pp. 282-287). IEEE, 2018.
[7] D. M. Truong, H. D. Nguyen, S. Vu, et al., Construct an intelligent querying system in education based on ontology integration. In 2022 IEEE International Conference on Computing (ICOCO), 340-345, Malaysia, 2022.
[8] B. Svendsen, & S. Kadry, A dataset for recognition of Norwegian sign language. International Journal of Mathematics, Statistics, and Computer Science, 2024, vol. 2.
[9] F. Kedwan, NLQ into SQL translation using computational linguistics. Journal of King Saud University-Computer and Information Sciences, 2022, vol. 34, no. 9, 6564-6582.
[10] Mahmoud Ismail, Naif El-Rashidy, Nabil M. Abdel-aziz, Mobile Cloud Database Security: Problems and Solutions, Journal of Fusion: Practice and Applications, Vol. 7 , No. 1 , (2022) : 15-29 (Doi : https://doi.org/10.54216/FPA.070102)
[11] G. Katsogiannis-Meimarakis, & G. Koutrika, A survey on deep learning approaches for Text-to-SQL. The VLDB Journal, 2023, vol. 32, 905 - 936.
[12] A.T. Nguyen, M.H. Dao, D. Nguyen, A Pilot Study of Text-to-SQL Semantic Parsing for Vietnamese. In Findings of the Association for Computational Linguistics: EMNLP 2020, 4079–4085, Online. Association for Computational Linguistics
[13] L. Wang, B Qin, B Hui, et al., Proton: Probing Schema Linking Information from Pre-trained Language Models for Text-to-SQL Parsing, Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2020), 1889 – 1898, 2022.
[14] A. Vaswani, N. Shazeer, N. Parmar, et al., Attention is all you need. Advances in neural information processing systems, 2017, vol. 30.
[15] K. Nassiri, & M. Akhloufi, Transformer models used for text-based question answering systems. Applied Intelligence, 2023, vol. 53, no. 9, 10602-10635.
[16] S. Grossberg, Recurrent neural networks. Scholarpedia, 2013, vol. 8, no. 2, 1888.
[17] A. Mastropaolo, S. Scalabrino, N. Cooper, et al., Studying the usage of text-to-text transfer transformer to support code-related tasks. In Proceedings of 43rd International Conference on Software Engineering (ICSE 2021), pp. 336-347. IEEE, 2021.
[18] M. H. Hwang, J. Shin, H. Seo, et al., Ensemble-NQG-T5: Ensemble Neural Question Generation Model Based on Text-to-Text Transfer Transformer. Applied Sciences, 2023, vol. 13, no. 2, 903.

Doi: https://doi.org/10.54216/FPA.150211
Received: August 21, 2023 Revised: December 17, 2023 Accepted: April 11, 2024

[19] V. Zhong, C. Xiong, & R. Socher, Seq2sql: Generating structured queries from natural language using reinforcement learning. Proc. of 6th International Conference of Learning Representation (ICLR 2018), Canada, 2018.
[20] Spider: https://yale-lily.github.io/spider (Access on 31 July 2023)
[21] T. Yu, R. Zhang, K. Yang, et al. Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018), pages 3911–3921, Brussels, Belgium, 2018.
[22] SYN dataset: https://github.com/ygan/Spider-Syn (Access on 31 July 2023)
[23] H. Nguyen, D. Truong, S. Vu, et al. Knowledge Management for Information Querying System in Education via the Combination of Rela-Ops Model and Knowledge Graph. J. Cases on Inf. Tech. (JCIT), 2023, vol. 25, no. 1, 13.
[24] M. Pham, K. Nguyen, V. T. Nguyen-Le, & H. Nguyen, An intelligent searching system for academic courses of programming based on Ontology Query-Onto. International Journal of Intelligent Systems Design and Computing (IJISDC). 2022, In press.
[25] B. Wang, R. Shin, X. Liu, O. Polozov, & M. Richardson, Rat-sql: Relation-aware schema encoding and linking for Text-to-SQL parsers. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020), pp. 7567–7578, 2020.
[26] N. V. Do, H. D. Nguyen, & A. Selamat, Knowledge-based model of expert systems using rela-model. International Journal of Software Engineering and Knowledge Engineering, 2018, vol. 28, no. 08, 1047-1090.
[27] T. Kudo, J. Richardson, SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP 2018), pp. 66–71, Brussels, Belgium, 2018.
[28] H.D. Nguyen, H. Huynh, T. Mai, et al., *Design an Ontology-based model for Intelligent Querying system in Mathematics Education*, Journal of Interdisciplinary Mathematics, 2023, vol. 26, no. 3, 449 – 473,
[29] D. Rothman, Transformers for Natural Language Processing: Build innovative deep neural network architectures for NLP with Python, PyTorch, TensorFlow, BERT, RoBERTa, and more. Packt Publishing Ltd., 2021.
[30] D. Almeida, C. Winter, J. Tang, & W. Zaremba, A generalizable approach to learning optimizers. CoRR abs/2106.00958, 2021.
[31] H. D. Nguyen, T. Huynh, S. N. Hoang, V. T. Pham, & I. Zelinka, Language-oriented Sentiment Analysis based on the Grammar Structure and Improved Self-attention Network. Proc. 15th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2020), pp. 339-346, 2020.
[32] H. Nguyen, V. Tran, T. Pham, A. T. Huynh, & N. V. Do, Ontology-based Integration of Knowledge Base for Building an Intelligent Searching Chatbot. Sensors & Materials, 2021, vol. 33, no. 9, 3101 – 3121.
[33] H. D. Nguyen, N. V. Do, N. P. Tran, & X. H. Pham, Criteria of a knowledge model for an intelligent problems solver in education. In Proceedings of 10th International Conference on Knowledge and Systems Engineering (KSE 2018), pp. 288-293, Ho Chi Minh, Vietnam, 2018.
[34] M. L. Gillenson, Fundamentals of database management systems. John Wiley & Sons, 2023.
[35] M. S. Baig, A., Imran, A. U. Yasin, et al., Natural language to SQL queries: A review. International Journal of Innovations in Science Technology, 2022, vol. 4, 147-162.
[36] Q. N. Naveed, M. R. N. Qureshi, N. Tairan, et al., Evaluating critical success factors in implementing E-learning system using multi-criteria decision-making. Plos one, 2020, vol. 15, no 5, e0231465.
[37] H. D. Nguyen, T. V. Tran, X. T. Pham, A. T. Huynh, V. T. Pham, & D. Nguyen, Design intelligent educational chatbot for information retrieval based on integrated knowledge bases. IAENG International Journal of Computer Science, 2022, vol. 49, no. 2, 531-541.

Doi: https://doi.org/10.54216/FPA.150211
Received: August 21, 2023 Revised: December 17, 2023 Accepted: April 11, 2024