WEP Warwick Evans Publishing

International Journal of Computer Science and Information Technology

ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 1, Number 1, Year 2023
DOI: https://doi.org/10.62051/ijcsit.v1n1.25
Journal homepage: https://wepub.org/index.php/IJCSIT/index

Intelligent Interaction of Fluorine containing materials Based on NL2SQL

Haoyu Liu, Yadong Wu*, Weihan Zhang

Sichuan University of Science and Engineering, Yibin, China
*Corresponding Author: Yadong Wu

ABSTRACT

The data on the production and development of fluorine containing materials are characterized by a large amount of data and a high degree of dimensionality of physical and chemical property characterization indicators. The manual way of analyzing the data item by item not only has high interaction cost, but also is difficult to analyze and explore the data intuitively. In order to efficiently utilize the data, this paper firstly constructs a dataset of fluorine-containing materials and proposes the Mengzi-ITPT model based on it, which takes Mengzi as the encoder and uses the attention mechanism to enhance the representation of the listed information. Meanwhile, for the data characteristics of fluorine containing materials, the training strategy of ITPT is adopted to improve the accuracy of the model. The experimental results show that the accuracy of the Mengzi-ITPT model query reaches 86.9% when the model is trained under the fluorine-containing material dataset.

KEYWORDS

NL2SQL; Fluorine containing materials; Semantic parsing; Supervised learning.

1. INTRODUCTION

With the rapid development of the fluorine chemical industry [1], a large amount of data has been generated in various aspects such as production and development. However, due to the large volume of data and the high dimensionality of physical and chemical property characterization indexes, the way researchers analyze the data one by one is not only costly, but also difficult to analyze and explore the data intuitively. If researchers are not familiar with the database language, it is difficult for them to accurately obtain the target data from the huge volume of data. For researchers who are familiar with database languages, it is also a tedious task to write a large number of accurate structured query languages for different scenarios.

NL2SQL is a transformation task that converts natural language into a structured query language that can be executed by a computer [2], and belongs to the semantic parsing subfield of natural language understanding [3]. Currently, there are two main approaches to the NL2SQL task: methods based on Seq2Seq [4] and methods based on sketch [5].Seq2Seq essentially treats the NL2SQL task as a translation task of a text, where the text is encoded by a CNN[6] or an LSTM [7] to obtain a semantic representation of the text, and a decoder is used to decode the text to generate the SQL statements. Although this method is well adapted to indeterminate long sequential sentences, it does not consider the syntax rules of SQL statements, so it leads to low accuracy of SQL statement generation. On the other hand, the method based on sketch treats the SQL statement as a fixed structure, and does not need to predict all the contents in the SQL statement, but only the key contents. This definitely greatly simplifies the difficulty of SQL statement prediction.

Content from this work may be used under the terms of CC BY-NC 4.0 licence (https://creativecommons.org/licenses/by-nc/4.0/).
Published by Warwick Evans Publishing.

In this paper, we propose a Mengzi-ITPT model by building a dataset of fluorine containing materials from scratch and based on the sketch method. The model uses Mengzi [8] as an encoder and enhances the representation of the column name information through the attention mechanism [9], and predicts the filling of the detail part in the SQL predefined touchpad through multiple subtasks. Meanwhile, for the data characteristics in the field of fluorine containing materials, the ITPT [10](withIn-Task Pre-Training)strategy is adopted to further train and optimize the pre-trained model to better adapt to the data distribution in the field of fluorine-containing materials, and test and validate it on the constructed fluorine-containing dataset. The results show that the model proposed in this paper is effective.

2. FLUORINE CONTAINING MATERIALS DATASET CONSTRUCTION

The sources of the table data in this paper are mainly obtained by researching a large number of materials such as specialized books, encyclopedic information resources, and expert knowledge. After collecting the tabular data, the following criteria are used to remove the ineligible tables by referring to the way WikiSQL [11] handles the data:

(1) Each row contains a different number of cells;

(2) The cell contains more than 50 words;

(3) The table header cell is empty;

(4) The table is less than 5 rows or 5 columns;

(5) More than 40% of the cells in a row contain the same content;

After deleting all the ineligible tables, 1492 table data remained. After that a simple data annotation platform is built which provides a complete view of the collected table data with column names, table contents and column data types such as numeric and text. After understanding the given table, two to three reasonable questions are asked based on that table and corresponding SQL statements are written. Figure 1 shows a brief description of the data annotation process.

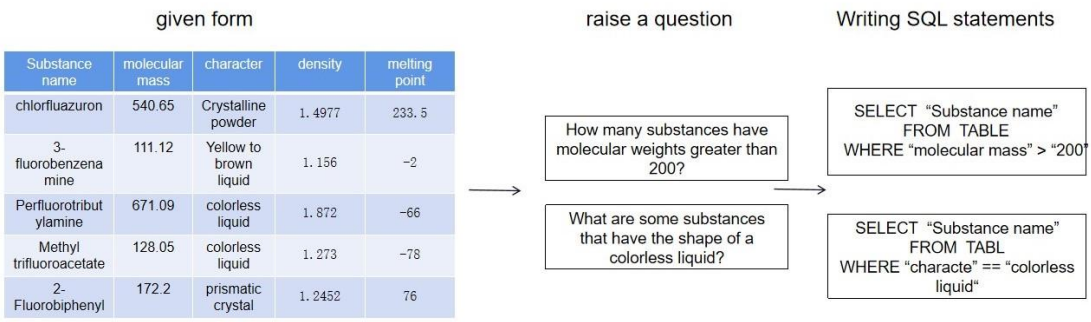

Figure 1. Data labeling process

A total of more than 4000 data were labeled for this experiment, on the basis of which the data were expanded using simBert [12] technique, and finally more than 12000 data were obtained. It is divided into training set, validation set and test set according to the ratio of 8:1:1. The content of the experimental training data is shown in Table 1. Where: question denotes the corresponding query; Table_id denotes the unique identifier of the table; "sql" denotes the correct SQL query statement corresponding to the question. Where "sel" that column is selected; agg that the selected columns of the aggregation operation, 0 to 5, respectively, that the selected columns do not operate, the average value (AVG), the maximum value (MAX), the minimum value (MIN), the number of counts (COUNT), the sum (SUM). cond_conn _op indicates the relationship between different conditional statements, 0 to 2 respectively means that there is only one WHERE conditional clause, multiple

conditional clauses are connected using AND and multiple conditional clauses are connected using OR." conds" denotes the triad in the where clause, i.e., (conditional column, conditional operator, conditional value), and the conditional operator uses 0 to 4 to denote greater than, less than, equal to, and not equal to.

Table 1. Sample data

data name	example
question	How many substances have molecular masss greater than 200
Table_id	“123456789abc”
sql	{"sel":[0],agg:[0],"cond_conn_op":0,"conds":[1,0,"200"]}

3. METHODOLOGY

3.1. Problem Definition

The purpose of the NL2SQL task is to convert natural language problems into machine-executable SQL query statements. Sketch-based approach is the current mainstream solution for NL2SQL because it can fully utilize the syntax rules of SQL statements.SQL statements have a fixed syntactic format, which is mainly composed of SELECT clauses and WHERE clauses in order, and its main structure is shown below in Figure 2, in which the blue part of the figure is the part that needs to be predicted.

```
SELECT    $AGG     $COLUMN
WHERE     $COLUMN  $OP    $VALUE
(AND      $COLUMN  $OP    $VALUE)*
```

Figure 2. SQL structure

3.2. Model Structure

In this paper, the model is based on slot value filling technique and the Mengzi-ITPT model is proposed. The model predicts the filling of the details in the SQL predefined touchpad through multiple subtasks. The structure of the model is shown in Figure 3 and contains three parts: the Mengzi encoder, the column name field representation, and the subtasks. Among them, the Mengzi encoder is responsible for encoding natural language questions and column name field information, and the column name field representation part uses an attention mechanism to enhance the column name field encoding, while referring to Typesql [13] to encode information from the database into the column name field. The subtasks use the encoded information as input to make predictions about the details in the SQL predefined touchpad.

In the sub-tasks section, there are four sub-tasks, namely S-sel-agg, W-conn-op, W-conds-ops W-conds-values, and the following is a detailed description of these sub-tasks:

S-sel-agg:selection field and aggregation operation prediction subtask, responsible for the prediction of selection fields and aggregation operations in the SELECT component, is a classification problem.

W-conn-op: comparison operation prediction subtask, responsible for the prediction of the relationship between multiple conditions of the WHERE component, is a classification problem.

W-conds-ops: conditional field and comparison operation prediction subtask, responsible for predicting conditional fields and their associated comparison operations in the WHERE component. It is a classification problem.

W-conds-values: the value extraction subtask, which extracts value values from natural language problems and combines them with the output of W-conds-ops to generate a ternary of final conditions.

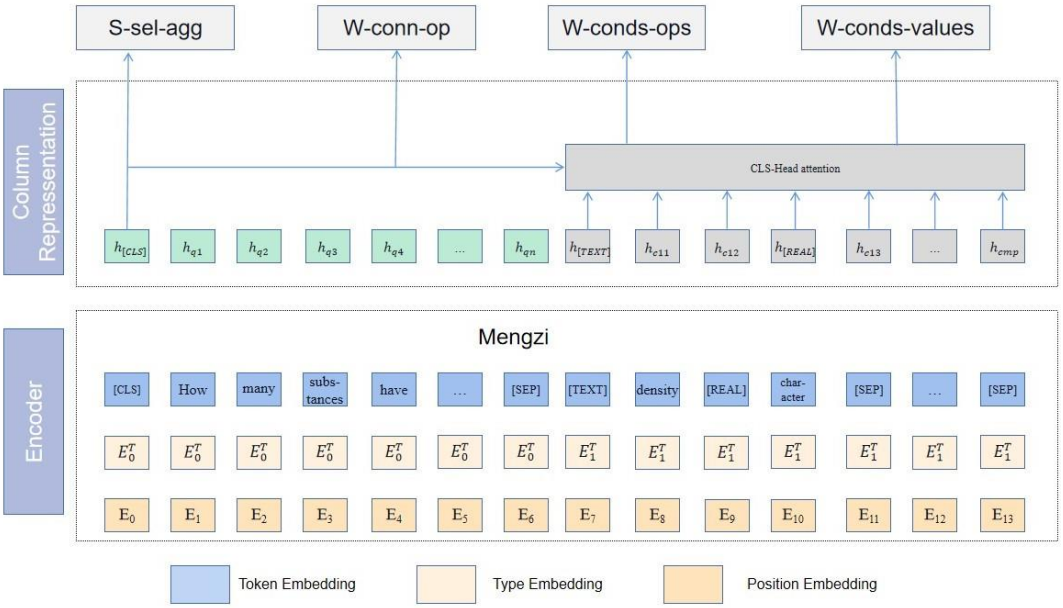

Figure 3. Model structure

3.2.1. Encoder

In the encoder, the query is concatenated with all the columns in the database and the different sentences are separated using the [SEP] tag. To further leverage the content in the database, the data attributes of the columns in the table are encoded using two special tags, each used to represent the type information of the columns. In this paper, the encoder uses Mengzi, which has the following advantages over Bert [14]:

(1) Linguistic features such as semantic roles and lexical annotations are fused into the Embedding representation to enhance the model's ability to model linguistic knowledge.

(2) The use of training correction strategy in the training strategy effectively improves the performance of the model on downstream tasks and helps the model to fight against attacks on synonym substitution.

The order of their input is as follows:

$$[CLS], q_1, q_2, \ldots, q_L, [SEP], [TEXT], \hbar_{11}, \hbar_{12} \ldots [SEP], \ldots, [REAL], \hbar_{n1}, \hbar_{n2}, \ldots, [SEP]. \quad (1)$$

Among them, $[CLS]$, $[SEP]$, $[TEXT]$and $[REAL]$ are special tags. $[CLS]$ is the start tag of the input sequence, which indicates the sequence information to some extent, and $[SEP]$ indicates the separator between the natural language question and the column name in the table. To further use the information in the database, the data format of the columns is encoded as well. Where $[TEXT]$denotes that all data in this column is of text type and $[REAL]$ denotes that all data in this column is of numeric type. q_i is the i-th token in the question. L is the length of the natural language question. n is the number of columns in the database, and H_{ni} denotes the i-th token in the n-th column.

The Mengzi encoder encodes the input information from three aspects, word, type and position, corresponding to three word embedding matrices. As shown in Fig. 3 above, Token embedding is responsible for encoding the word information; Type embedding is responsible for encoding the type information of the input sequence, where E_0^T represents the input natural language question and E_1^T represents the column name information of the table; and Position embedding is responsible for encoding the position information of the input sequence. The vectors encoded by Mengzi are obtained by fusing these three types of word embedding information. Its output is of the form:

$$h_{[CLS]}, h_{q1}, h_{q2}, \ldots, h_{qL}, h_{[SEP]}, h_{[TEXT]}, h_{c11}, h_{c12}, \ldots, h_{[SEP]}, \ldots, h_{[REAL]}, h_{c21}, h_{c22}, \ldots, h_{[SEP]}. \quad (2)$$

Where $h_{[CLS]}$, $h_{[SEP]}$, $h_{[TEXT]}$ and $h_{[REAL]}$ denote the encoding vectors of $[CLS]$, $[SEP]$, $[TEXT]$ and $[REAL]$ tokens respectively, with encoding vector dimensions d, h_{qt} denotes the encoding vector of the token of the t-th token in the sequence of the natural language problem, and h_{cij} denotes the encoding vector of the token of the j-th token in the sequence of the i-th field.

3.2.2. Column Representation

After obtaining the context encoded information of the natural language problem and the column name field as well as the attribute information of the columns after using the Mengzi encoder. The context encoded information of the column name field is further represented intensively using the attention mechanism and the overall encoded information of the input sequence $h_{[CLS]}$.Regarding the attention weights of the context encoded information h_{cij} of the column name field and the overall encoded information $h_{[CLS]}$ of the input sequence, they are calculated as follows:

$$s_{ij} = dot\left(Uh_{[CLS]}, Vh_{cij}\right). \quad (3)$$

$$a_{ij} = \frac{exp(s_{ij})}{\sum_{k=1}^{m} exp(s_{ik})}. \quad (4)$$

Where, $U$and V are the parameters that can be learned, $U, V$are $d \times d$dimensional, dot denotes the dot product operation. s_{ij} denotes the similarity of the encoded information h_{cij} of the j-th token in the sequence of the i-th column name field to $h_{[CLS]}$, and a_{ij} denotes the weight of attention after normalization process. The representation of the column name field based on the attention mechanism is computed as shown below:

$$r_{ci} = \sum_{j=1}^{m} a_{ij} \cdot h_{cij}. \quad (5)$$

$$f_{ci} = r_{ci} + h_{[CLS]}. \quad (6)$$

Where r_{ci} is the i-th column name field representation based on the attention mechanism. In order to further utilize the overall coding information $h_{[CLS]}$, r_{ci} is fused with $h_{[CLS]}$ to better align the natural language problem and the column name field, and to improve the prediction ability of the column name field. In this paper, we use f_{ci} as the i-th column name field representation as the input for subsequent subtasks.

3.2.3.Subtask Output

The model in this paper uses four subtasks to populate the predictions for the details in the SQL predefined templates, which are shown in Figure 2, where the blue parts indicate the parts to be populated. Each of these subtasks is described next:

S-sel-agg is used to predict which column in the clause is selected and what aggregation operation is to be performed. The prediction set is [0,1,2,3,4,5,6], which is modeled as a seven classification problem in this paper. Where 0-5 represent no operation on the selected column, average value (AVG), maximum value (MAX), minimum value (MIN), count (COUNT), sum (SUM) operation, respectively, and 6 is a new category, which represents NO_OP, i.e., the column is not selected. For example, if a column is selected and the MAX aggregation function needs to be done, the column is set to 2. The rest of the columns are all set to 6, which means that the column is not selected and no aggregation operation needs to be done S-sel-agg uses the overall coded information of the input sequence, $h_{[CLS]}$, as an input to make a prediction, which is computed as shown below:

$$p_1 = softmax(W_1 h_{[CLS]}). \tag{7}$$

Where p_1 denotes the probability that S-sel-agg outputs on the prediction set and W_1 is a learnable parameter.

W-conn-op is used to predict the join operator between different conditional triples in WHERE clauses with a prediction set of [0,1,2]. In this paper, it is modeled as a triple classification problem. For example, "what are the materials with density greater than 1 and less than 2", the connection operator is 1 for AND. W-conn-op uses the overall coded information of the input sequence, $h_{[CLS]}$, as input to make a prediction. to make a prediction, which is computed as shown below:

$$p_2 = softmax(W_2 h_{[CLS]}). \tag{8}$$

Similar to p_1 , p_2 denotes the probability that W-conn-op outputs on the prediction set and W_2 is a learnable parameter.

W-conds-ops is used to predict the selected conditional columns in the conditional triad in the WHERE clause and their concatenation relativities with the conditional values. The prediction set is [0,1,2,3,4]. Where 0-3 are greater than, less than, equal to and not equal to, respectively, a new column NO_OP, used to indicate whether the column is selected or not, similar to the operation in the S-sel-agg. For example, if a column is selected and the conditional join relation is equal, then this column is set to 2 and all the rest of the columns are set to 4, indicating that the rest of the columns are not selected.W-conds-ops uses the f_{ci} input for prediction, which is computed as shown below:

$$p_3 = softmax(W_3 f_{ci}). \tag{9}$$

Where p_3 denotes the output probability of making a prediction in the i-th column name field, where W_3 is a learnable parameter.

W-conds-value is mainly used to predict the conditional values in WHERE clauses with a prediction set of [0,1]. Based on the predicted conditional columns and conditional connectives in W-conds-ops, enumerate generates possible combinations of (cond_col,cond_op,cond_val) and combines them with

the information from f_{ci} to determine whether the candidate conditional combinations are correct. The computation is shown below:

$$p_4 = sigmoid(W_4 f_{ci}). \tag{10}$$

where p_4 denotes the probability of correct prediction and W_4 is the parameter available for learning.

3.3. WithIn-Task Pre-Training

Since the Mengzi pre-training model is trained on data from the generalized domain, the distribution of these data may be different from that of the target domain. Therefore, if the Mengzi pre-training model is directly applied to the target domain, the optimal performance may not be obtained. To solve this problem, this paper refers to the withIn-Task Pre-Training proposed by Sun [10] et al. to further pre-train the Mengzi model using data from the target domain. In this way the model can be better adapted to the data distribution of the target domain and thus achieve better performance on the target task.

4. EXPERIMENT

4.1. Experimental platform and evaluation indicators

The experimental platform used in this experiment is equipped with NVDIA Quadro RTX 6000, the deep learning framework uses tensorflow, and the experiment as a whole is realized in python language.

The model is evaluated in the following two ways: executive form accuracy and logical form accuracy[15]. The formulas are shown below.

$$Accuracy_{ex} = \frac{N_{ex}}{N}. \tag{11}$$

$$Accuracy_{lf} = \frac{N_{lf}}{N}. \tag{12}$$

Execution form accuracy is a more intuitive method of evaluating a generated SQL query statement by executing it on a database and then comparing the results of the query with the actual results. If the query results of the generated SQL statement and the actual results are identical, then the SQL query statement is considered correct. The advantage of this approach is that it is intuitive and easy to understand, and can capture more diverse SQL query statements.

Logical Formal Accuracy is a more rigorous evaluation method that directly compares every SELECT clause and WHERE clause in the generated SQL query statement and the real SQL query statement, and considers the generated SQL query statement to be correct only if they are identical. This approach has the advantage of catching more errors, but has the disadvantage of potentially ignoring SQL query statements that are semantically equivalent but formally different.

In real scenarios, since different SQL query statements may produce the same results, the general execution form accuracy rate will be higher than the logical form accuracy rate. In order to be able to accurately reflect the accuracy of the model for generating SQL query statements and to integrate the advantages of execution form accuracy and logical form accuracy, this paper uses execution form accuracy, logical form accuracy and average execution accuracy. The average implementation accuracy is shown below.

$$Accuracy_{mean} = \frac{Accuracy_{ex} + Accuracy_{lf}}{2}. \quad (13)$$

4.2. Baseline Model

In this paper, we use SQLNet [16], SQLova [17], and X-SQL [18] as baseline comparison models, and these three baseline models are described below.

SQLNet is the first model based on the slot-value filling technique and lays the foundation for subsequent slot-value filling class models.

SQLova is the first model to introduce BERT pre-training technique in SQL parsing task, which is a combination of SQLNet and BERT.

X-SQL uses "CONTEXT REINFORCING LAYER" to enhance the representation of field information in the database.

4.3. Experimental results and analysis

4.3.1. Model comparison experiments and analysis of results

The results of this experiment are shown in Table 2 below:

Table 2. Model comparison results

Model	Dev LF[%]	Dev X[%]	Dev MX[%]	Test LF[%]	Test X[%]	Test MX[%]
SQLnet	62.2	65.7	63.9	61.2	64.4	62.8
SQLova	81.2	85.4	83.3	80.9	83.7	82.3
X-SQL	83.1	86.7	84.9	82.8	86.4	84.6
ourModel	85.8	88.1	86.9	84.7	87.8	86.3

Table 2 shows the experimental results of the model proposed in this paper and other baseline models on the fluorinated material dataset, and the experimental results show that the model proposed in this paper achieves the optimal performance on the fluorinated material dataset. In terms of logical form accuracy, execution form accuracy, and composite accuracy, it reaches 84.7%, 87.8%, and 86.3%, respectively, and achieves an accuracy improvement of 1.9%, 1.4%, and 1.7% compared to the X-SQL model.

4.3.2 Ablation experiments

In order to further explore the improvements made in this paper, another ablation experiment was conducted and the results are shown in Table 3 below:

Table 3. Results of ablation experiments

Model	Test LF[%]	Test X[%]	Test MX[%]
ourModel	84.7	87.8	86.3
-ITPT	83.9	85.8	84.9
-Mengzi+BERT-base	82.9	84.7	83.8

The ablation experiments show that since Chinese words contain more semantic information, the use of the Mengzi pre-training model can get better results, improving the overall accuracy by 1.1% over the use of the BERT pre-training model. Moreover, due to the use of ITPT strategy, the data

distribution of the pre-trained model can be made as consistent as possible with the dataset of the target domain so as to improve the accuracy of the model, which is 1.4% higher than that of the BERT pre-trained model in terms of the overall accuracy.

5. CONCLUSION

In this paper, to address the problem of slow query speed and high cost of fluorine containing material data, we build a fluorine containing material dataset from zero, propose the Mengzi-ITPT model based on the sketch, and use the attention mechanism to enhance the representation of the column name information.Meanwhile, according to the characteristics of fluorine containing material dataset, the training strategy of ITPT is adopted, and the experimental results show that the model proposed in this paper is simple and effective. In the subsequent research work, the accuracy and generalization ability of the model algorithm can be improved by obtaining more fluorine-containing material data and combining with the knowledge graph related technology to further understand the semantic information in natural language problems.

REFERENCES

[1] Wang, Y. Opportunities and Prospects of Fluorocarbon Application in New Energy Field[J]. new materials industry,2019,(10):3034.DOI:10.19599/j.issn.1008-892x.2019.10.008.

[2] Xiaoyu Z ,Fengjing Y ,Guojie M , et al. M-SQL: Multi-Task Representation Learning for Single-Table Text2sql Generation[J]. IEEE Access,2020,8.

[3] Zhang, Z., Wang, B., Zhao, J., et al. Intelligent Interaction of Power Data Based on NL2SQL[J].Grid technology,2022,46(07):2564-2571.DOI:10.13335/j.1000-3673.pst.2021.1311.

[4] Liu T, Wang K, Sha L, et al. Table-to-text generation by structure-aware seq2seq learning[C]//Proceedings of the AAAI conference on artificial intelligence. 2018, 32(1).

[5] Dong L, Lapata M. Coarse-to-fine decoding for neural semantic parsing[J]. arxiv preprint arxiv:1805.04793, 2018.

[6] Chua L O. CNN: A paradigm for complexity[M]. World Scientific, 1998.

[7] Memory L S T. Long short-term memory[J]. Neural computation, 2010, 9(8): 1735-1780.

[8] Zhang Z, Zhang H, Chen K, et al. Mengzi: Towards lightweight yet ingenious pre-trained models for chinese[J]. arxiv preprint arxiv:2110.06696, 2021.

[9] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[10] Sun C, Qiu X, Xu Y, et al. How to fine-tune bert for text classification?[C]//Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings 18. Springer International Publishing, 2019: 194-206.

[11] Zhong V, Xiong C, Socher R. Seq2sql: Generating structured queries from natural language using reinforcement learning[J]. arxiv preprint arxiv:1709.00103, 2017.

[12] Su J. Simbert: Integrating retrieval and generation into bert[J]. Tech. Rep, 2020.

[13] Yu T, Li Z, Zhang Z, et al. Typesql: Knowledge-based type-aware neural text-to-sql generation[J]. arxiv preprint arxiv:1804.09769, 2018.

[14] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arxiv preprint arxiv:1810.04805, 2018.

[15] Chang S, Liu P, Tang Y, et al. Zero-shot text-to-SQL learning with auxiliary task[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34(05): 7488-7495.

[16] Xu X, Liu C, Song D. Sqlnet: Generating structured queries from natural language without reinforcement learning[J]. arxiv preprint arxiv:1711.04436, 2017.

[17] Hwang W, Yim J, Park S, et al. A comprehensive exploration on wikisql with table-aware word contextualization[J]. arxiv preprint arxiv:1902.01069, 2019.

[18] He P, Mao Y, Chakrabarti K, et al. X-SQL: reinforce schema representation with context[J]. arxiv preprint arxiv:1908.08113, 2019.