Title	A study on transforming natural language sentence to SQL queries and its application for low-resource languages
Author(s)	PHAM, VIET CUONG
Citation	
Issue Date	2022-12
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/18164
Rights	
Description	Supervisor:NGUYEN, Le Minh, 先端科学技術研究科, 修士(情報科学)

Master's Thesis

A study on transforming natural language sentences to SQL queries and its application for low resource languages

PHAM, Cuong Viet

Supervisor NGUYEN, Minh Le

Graduate School of Information Science
Japan Advanced Institute of Science and Technology

12, 2022

Abstract of Master's Thesis

With a population of over 90 million people, Vietnam is an attractive market for many domestic and foreign investors in Southeast Asia. Currently, companies and corporations in Vietnam are deploying and implementing management by intelligent digital technology, in which the exploitation and use of relational database management systems have been applied for years since the beginning of the year. Therefore, at present, the text-to-SQL problem is also an essential need for many foreign investors who want to exploit and learn about their market in Vietnam and manage the company's domestic managers. I researched and learned effective solutions to solve this problem in my research. Due to the short research time, I propose a test of a solution that has been applied in English, the RAT SQL method. This is a solution that many researchers in this field have exploited and improved in the English language and achieved high efficiency. Compared with previous methods, RATSQL has focused on expanding the exploitation of the relationship between the question and the cell value in the database. Using a related aware mechanism has helped to find the implicit connection between the query word and the elements in the database schema. During the experiment with RATSQL, I adjusted the basic parts in RATSQL to match the characteristics of the Vietnamese language, and translated 166 databases in the Spider dataset. The results obtained for the highest accuracy are 64.7% with the RATSQL method combined with PhoBert pre-train.

Keywords:

Text-to-sql, Natural-language understanding, Semantic parsing

Contents

This thesis was prepared according to the curriculum for the Collaborative Education Program organized by Japan Advanced Institute of Science and Technology and Le Quy Don Technical University .

List of Figures

List of Tables

Chapter 1

Introduction

1.1. Semantic Parsing

Nowadays, chatting with robots, smart speakers, or virtual assistants (also known as intelligent assistants) like Google Assistant or Alexa is becoming increasingly popular. To create a product like them, IT engineers need to know about natural language processing. And an important problem in the field of natural language processing is semantic parsing. There are many definitions of semantic parsing, and I give a concept of semantic parsing as follows.

Semantic Parsing is the task mapping from a natural sentence into the logical representation which is widely applied in realistic applications. The key task of semantic parsing is to find a function such that:

$$f : Sentence- > LogicForm$$

Three factors need to be taken into regard by a semantic parser:

- Factor modelling, how to represent a logic form
- Factor parsing, design a grammar and parsing algorithm
- Factor learning, use supervision to fix parameters

1.2. Text-to-SQL

Text-to-SQL is an essential problem in semantic parsing that has been interesting and studied by many researchers recently. Before learning about this problem, I first introduced the structured query language.

Structured query language such as SQL is a computer language used to execute query commands from an object-relationship database management system such as add, edit, delete, etc. In 1987, SQL was officially published by the American National Standards Institute (ANSI). Since then, database management systems using SQL have been increasing and storing a vast amount of human knowledge. Highlights are data from the financial, e-commerce, and pharmaceutical industries.

SQL and Relational Database Management Systems(RDBMS) have been around for a long time, time-tested, and heavily used. They are designed for reliable transactions and particular queries in applications. However, SQL databases are not without limitations and are not always suitable for all data storage and retrieval needs.

NoSQL databases were created to overcome these constraints. Data is stored and managed by the NoSQL database system in a way that promotes rapid operation and broad flexibility. Businesses like Google, Amazon, Yahoo, and Facebook created many NoSQL database systems to find better ways to handle or store data for massive websites. In contrast to SQL databases, many NoSQL databases may be deployed horizontally across hundreds or thousands of servers.

Despite the explosion of NoSQL in recent years, SQL is still making a comeback to become the favored interface for data analysis. Again, SQL is listed as one of the top skills to master in the future. Indeed, according to the latest statistics on the popularity of database management systems of StackOverflow 2022 [1], we see that database management systems using SQL language are always more and more popular than other database management systems - Database administrators do not use SQL query language. Specifically, MySQL, PostgreSQL, and SQLite management systems are the management systems using SQL, ranked 1,2,3. In contrast, the MongoDB management system is the management system that specializes in using nonSQL queries, ranked 4th.

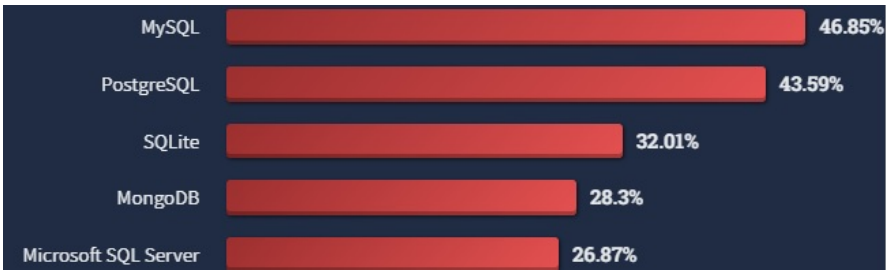


Figure 1.1: Most Popular Databases – Source: StackOverflow Developer Survey Results 2022

With a long history of establishment and use by many RDBMSs, databases using SQL query language have become a large knowledge store that many people are interested in exploiting. An exploit that is of interest to many researchers is text to SQL. And that's an essential reason for the birth of the text-to-SQL problem.

Text-to-SQL, which is a task to translate the natural language utterances into SQL queries automatically, has just been arising due to the appearance of the Spider data set.

- Input: user's spoken or text in natural language
- Output: SQL queries

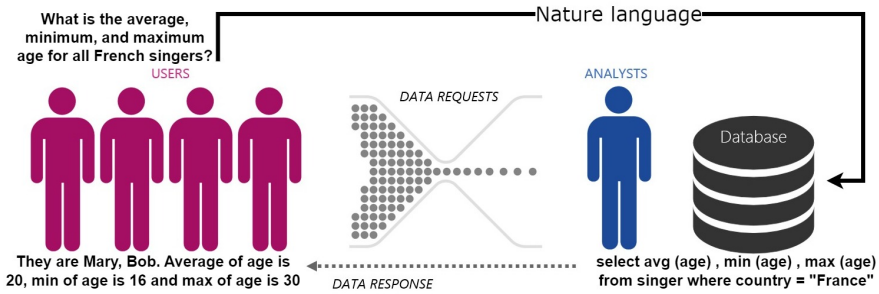


Figure 1.2: Illustrate the process of translating the question into SQL and getting the results

1.3. Dataset

Currently, text-to-SQL is still a challenging problem that many universities and technology corporations research and test before putting into practice. Therefore, they also provide many data sets to test and evaluate the methods and feasibility of the project. According to the survey of Ayush Kumar et al in 2022 [7] , there are about some datasets for the problem of researching and evaluating text-to-SQL solutions.

- ATIS (Air Travel Information System) / GeoQuery is the data set related to audio recordings and hand transcripts of individuals utilizing automated travel inquiry systems seeking information about flights.
- IMDB is a dataset for natural language processing with more than 50,000 movie reviews
- Advising The author was Finegan-Dollak et al. in 2018. The Advising dataset consists of 457 JSON files related to course information questions at the University of Michigan.
- MAS Microsoft Academic Search is a related database academic and social networking and a dataset of queries
- WikiSQL consists of a dataset of 87,726 SQL queries and pairs of relevant questions. The training dataset is 61,297 examples, the development dataset is 9,145, and the test dataset is 17,284.
- Spider 1.0 is Yale university's dataset for English language. 1. Large: over 10,000 questions and 6,000 quieries for English language; 2.Complex: SQL quieries consist of select, where, groupby, orderby,having, and forein key. 3. Cross-domain: because of 200 complex databases for English language.

Dataset	# Q	# SQL	# DB	# Domain	# Table /DB	ORDER BY	GROUP BY	NESTED	HAVING
ATIS	5,280	947	1	1	32	0	5	315	0
GeoQuery	877	247	1	1	6	20	46	167	9
Scholar	817	193	1	1	7	75	100	7	20
Academic	196	185	1	1	15	23	40	7	18
IMDB	131	89	1	1	16	10	6	1	0
Yelp	128	110	1	1	7	18	21	0	4
Advising	3,898	208	1	1	10	15	9	22	0
Restaurants	378	378	1	1	3	0	0	4	0
WikiSQL	80,654	77,840	26,521	-	1	0	0	0	0
Spider	10,181	5,693	200	138	5.1	1335	1491	844	388

Figure 1.3: Detail datasets

Below is a drawing depicting the complexity of the queries of each dataset compared to the Spider dataset. This show that the Spider dataset is a more complex and multi-domain semantic analysis than other data sets.

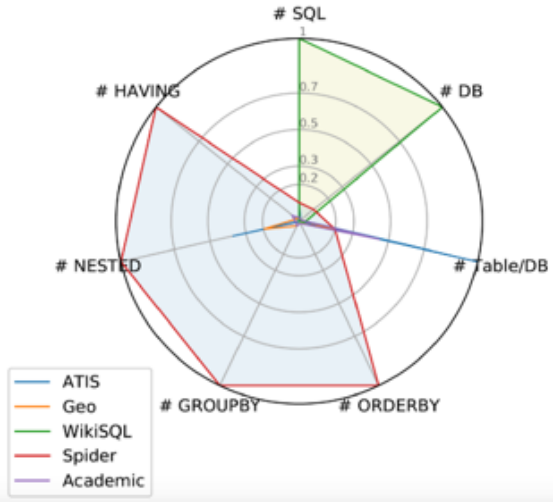


Figure 1.4: Comparing Spider with the previous datasets

This problem was also exciting and studied by many Vietnamese researchers very early. In 2020, almost the entire English Spider dataset was translated into Vietnamese by the Dat et al.2020, except for the test set and a small part of the train set. Because the English test set was deployed on Spider's server, the author team has trained to translate the train set and dev set into Vietnamese. In my research, I focus on researching and experimenting with the dev set because the

author mainly analyze and evaluate this data set.

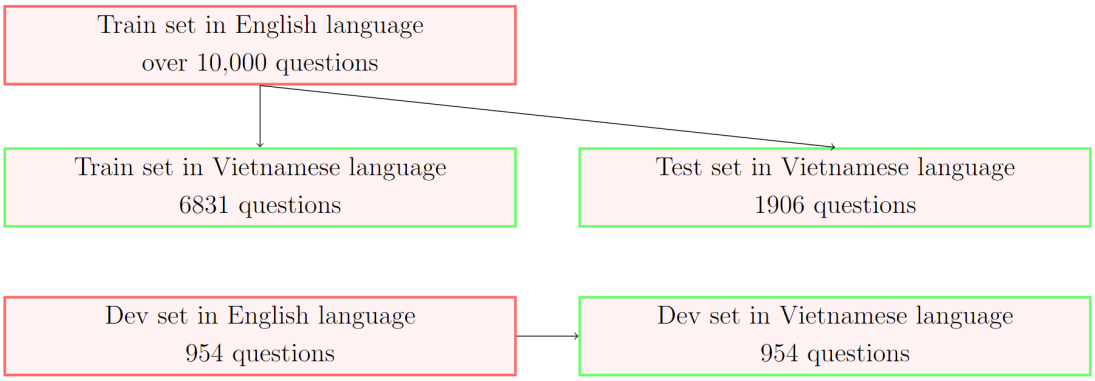


Figure 1.5: Split the English Spider set data to translate into Vietnamese

1.4. Challenges

When translating natural language questions (both English and low-resource language as the Vietnamese language) into SQL queries to answer questions from a database, contemporary semantic parsing models struggle to solve some important problems:

- Create a representation of the schema and facilitate to decode a SQL query. (Schema encoding)
- Detecting tables or columns that mention in utterances (Schema linking)
- Infer columns mention from cell values (Schema encoding, Schema linking)
- Compose complex SQL queries

The following are three examples illustrating such challenges in the English language taken from Wang's article [19].

- Pain Point 1: Fail to match and detect the column mentions. Which professionals live in a city containing the substring 'West'? List his or her role, street, city and state.
  SELECT role code, street, state FROM Professionals WHERE city LIKE '%West%'
  Missing column city in SELECT clause.

- Pain Point 2: Fail to infer columns based on cell values. Give the average life expectancy for countries in Africa which are republics?
  SELECT Avg(LifeExpectancy) FROM country WHERE Continent = 'Africa'
  Missing GovernmentForm = 'Republic'.

- Pain Point 3: Fail to compose complex target SQL. Which semesters do not have any student enrolled? List the semester name.
  SELECT semester_name FROM Semesters WHERE semester_id NOT IN (SELECT semester_name FROM Student Enrolment)
  Should use semester_id in nested SQL to align with the column in WHERE clause.

Figure 1.6: Example about Challenges

In addition to the above challenges, with other languages like Vietnamese, when dealing with text-to-SQL problems, there are other challenges such as:

- Some low-resource languages such as Myanmar, Vietnamese, etc , almost database designers use English language for table/ column names, They use a description in Vietnamese for each column and the corresponding table.

- Unlike English, Thang et al., 2008 showed that 85% of word types in Vietnamese are composed of two or three syllables, even four syllables [5]. That means that one English word can be translated into many Vietnamese words, which increases the complexity of the processing. So in Vietnamese Vin Spider dataset, authors used '_' to connect syllables of word such as Bao_nhiêu, quốc_gia, châu_Âu,etc. To solve this issue in case of using Bert multilingual-base pre-train, I removed all '_' of each word.

- In addition, when translating from English to Vietnamese, there will also be some problems, such as redundant translation, the English part does not convey all layers of meaning in Vietnamese, and Problems with vocabulary selection (words and structures).

Notes

Most databases that have many columns and tables are added into the test set.

Chapter 2

Related Works

2.1. Current approaches

With the challenges mentioned above, researchers in natural language processing have proposed many approaches to the text-to-SQL problem and have obtained many remarkable results.

EditSQL [24] includes three components (1) Utterance-Table Encoder to encode the utterance explicitly and table schema by a BiLSTM or BERT embedding (2) Interaction Encoder with Turn Attention to incorporate the recent utterance history by a BiLSTM, and (3)Table-aware Decoder with attention, based on the outputs of both encoders to generate a SQL query by using an LSTM.

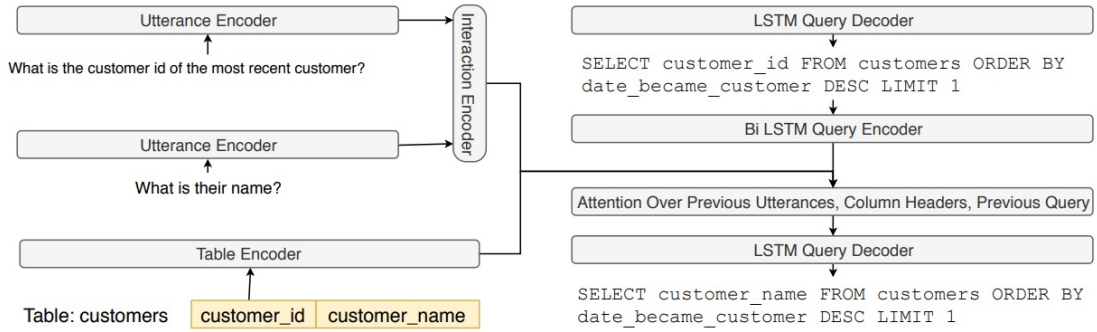


Figure 2.1: Model architecture Edit SQL. (source in [24])

IRNet uses the schema linking in Schema Encoder component to detect the columns and the tables occurred in a utterence. In the decoder component, it uses an abstract syntax tree SemQL query synthesizes a database schema and the results of schema linking by using a BiLSTM-based utterence encoder and an attention-based schema encoder together with a grammar-based LSTM decoder.

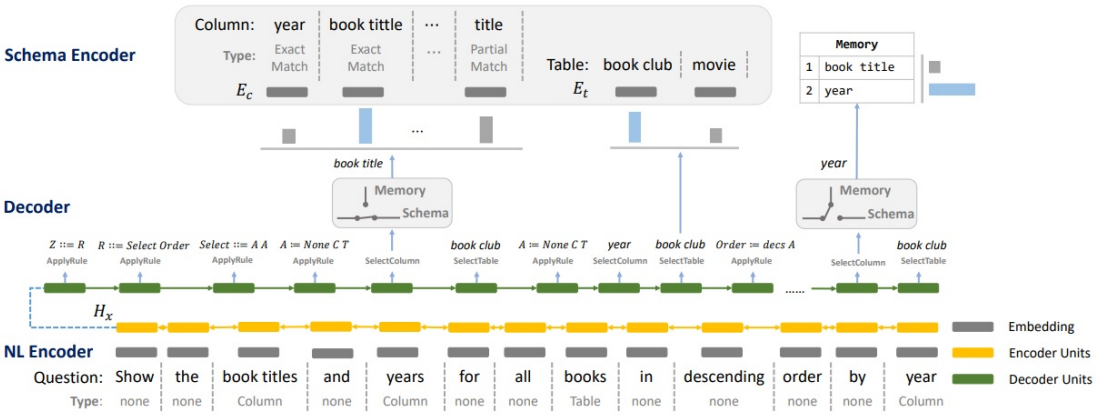


Figure 2.2: Model architecture IRNet. (source in [6])

RATSQL uses relation-aware self-attention and considers both direct relation (schema) and implicit relation (linking between question and schema) in encoding part, that improves the representation ability of the model.

RATSQL+ GAP [16] Peng Shi et al. proposed a novel framework by crawling large data SQL queries on github site for pre-training semantic parsers to exploits multiple pre-training tasks and synthetic data. By utilizing SQL-to-Text and Table-To-Text generative models to provide synthetic data for learning combined representations of textual data and table schema, they offered a novel approach to overcoming pre-training data difficulties.

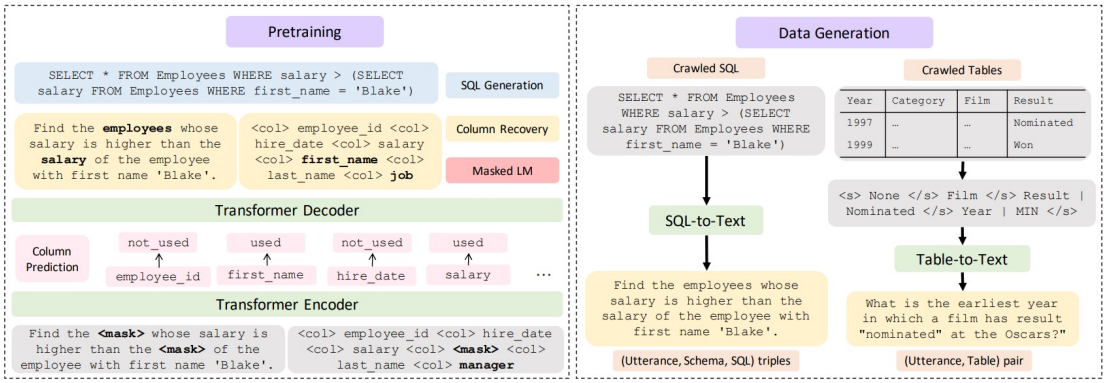


Figure 2.3: Model architecture RATSQL + GAP.

BERT [4], which stands for "Bidirectional Encoder Representations from Transformers", is a pre-trained model in the area of natural language processing that learns two-dimensional contextual representation vectors of words that are used to transfer to issues. Through its context, BERT has succeeded in building on recent work in locating representations of words in digital space (a realm that computers can understand).

mBert [13] Google Research released a multilingual version of BERT with over 100 language support includes Vietnamese shortly after introducing BERT.

PhoBert This is a pre-trained monolingual language, only training for Vietnamese. The training is based on Facebook's RoBERTa-like approach and architecture introduced by Facebook in mid-2019. This is an improvement over the previous BERT.

BART [8] is a model introduced by Facebook AI, a new pre-trained model that combines the advantages of BERT and GPT. The strength of BERT lies in capturing two-dimensional context, while GPT is self-regressive. With the advent of BART, text generation and comprehension tasks can be performed with the same model.

BARTpho [11] The first publicly available large-scale monolingual sequence-to-sequence models pre-trained for Vietnamese are two BARTpho variants, BARTpho-syllable and BARTpho-word. BARTpho, which is especially suited for generative NLP problems, leverages the "big" architecture and pre-training method of the BART sequence-to-sequence denoising model.

A novel pre-train method called "efficiently learning an encoder that classifies

token replacements accurately" (ELECTRA) [3], was unveiled by Google AI. It outperforms existing methods using the same computational resources.

Among the many approaches to problem-solving, I list some methods that give significant results along with the corresponding applied languages in Table 2.1.

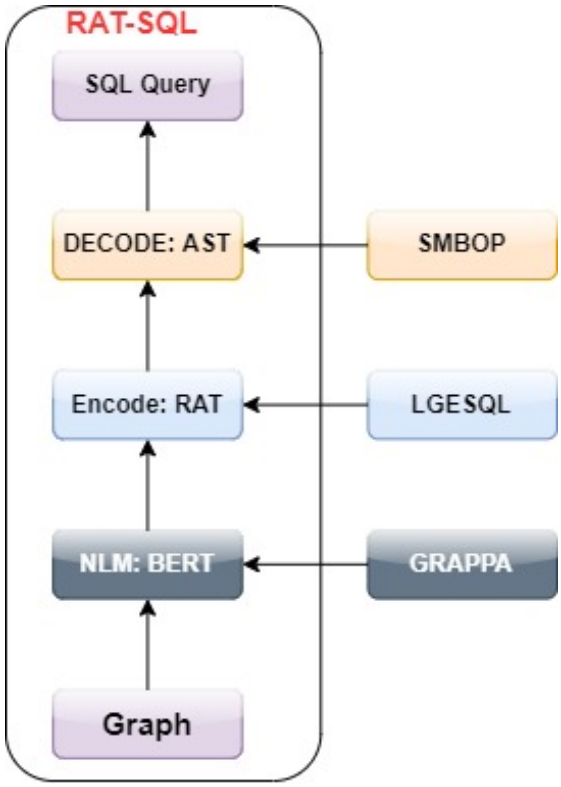
Approach	Language	Exact
IRNet+BERT_large [6]	English	54.7%
RATSQL+BERT_large [19] [20]	English	65.6%
BRIDGE v2+BERT [9]	English	67.5%
RATSQL+ SmBop + Grappa [14]	English	71.1%
RATSQL+ GraPPa + GP [22]	English	73.4%
LGESQL+ELECTRA [2]	English	75.1%
ISESL-SQL + ELECTRA_large [10]	English	75.8%
RAT-SQL (without schema linking) + Multilingual BERT [1]	Chinese	41.4%
LGESQL + GTL + Electra + QT [18] [1]	Chinese	64.4 %
EditSQL [12] [17]	Vietnamese	33.7%
EditSQL(DeP) [12] [17]	Vietnamese	45.3%
EditSQL+PhoBERT_large [12] [17]	Vietnamese	56.7%
IRNET(DeP) [12] [17]	Vietnamese	52.2%
IRNET+PHOBERT_large [12] [17]	Vietnamese	60.2%

[1] `https://taolusi.github.io/CSpider-explorer/` .

Table 2.1: Results on Spider dev set

2.2. RAT SQL model

In the above approaches, RATSQL has proven to be better at problem-solving than other methods, such as Schema encoder IRNET does not exploit schema relations fully. And the test results on the dev set and the English test proved it. With that proof, currently, on the Spider dataset for Vietnamese, the current approach to solving the problem only stops at the IRNET + BERT method, giving the highest test result is 60.2% on the dev set.



- **RAT-SQL** : relation-aware schema encoding and linking
- **SMBOP** : semi-autoregressive bottom-up
- **LGESQL** : line graph enhanced Text-to-SQL Model
- **GRAPPA** : grammar-augmented pre-training for table semantic parsing

Table 2.2: RATSQL model and its improvement

Although RATSQL has proven its effectiveness on the English dataset, when applied to the Vietnamese Spider set, it will still face many other challenges due to the language difference presented in Section 1.3. In addition, other improved methods of RATSQL, such as SMBOP, GRAPPA are not suitable for processing in the Vietnamese language.

Chapter 3

Proposed Model

3.1. Implementation Approache

Since its publication in 2020, RATSQL has been exploited, improved, and applied by many researchers in their approaches to obtain significant results with the English language. However, in my research, due to the short time, I suggest testing on the original RATSQL version and enhancing by mBert, phoBERT, BARTPho pre-train.

I propose the adjustment RAT SQL framework for the Vietnamese language, a method uses the relation-aware self-attention mechanism and solve challenges concern schema encoding, schema linking, and feature representation within a text-to-SQL encoder.

Unlike the previous methods, RATSQL is a method to exploit the relationship between questions and cell values in the database. This is suitable because most companies, corporations, and the Vietnamese government store data in Vietnamese in the database. In addition, RAT-SQL uses the relate-aware mechanism, which enables a more robust discovery of the relationship between the question and the database schema.

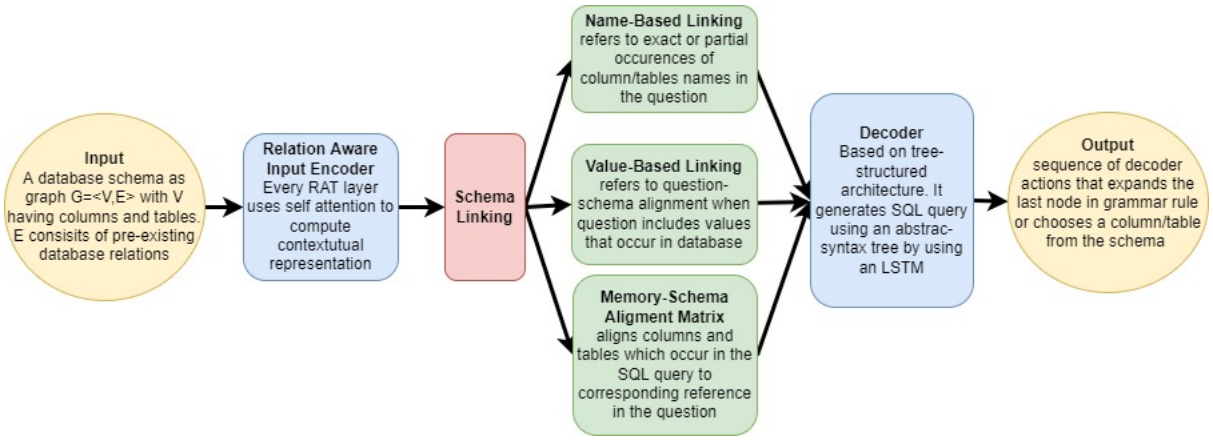


Figure 3.1: Original RATSQL model

3.2. Details of components

Before analyzing each part of RATSQL, I generalize the problem as follows: **Given:** Natural language utterence Q and the relational database that has a schema S=<C,T> with C and T are the set of columns and tables describe in Vietnamese language.

Each coulumn c_i belong C consists of words $c_{i,1}, c_{i,2}, ..., c_{i,n}$.

Each table t_i belong T consists of $t_{i,1}, t_{i,2}, ..., t_{i,m}$.

n and m are the length of C and the length of T.

Result: Generate a corresponding SQL query P

(1) Input:

Firstly, I build a one-to-one mapping between the table names and columns in English with their respective descriptions in Vietnamese.

Secondly, I define a database schema as the graph G=<V,E>. Construct Graph G is a directed graph where each node and edge has a label.

Set of vertices V consists of two parts: column describes, table describes. For column, add column type to vertex label at the same time.

Set of edges E in G graph consists of three parts: the first is the table connection relationship defined by the Database Schema, the inclusion relationship between table and column, etc. that show in figure 3.2.

Type of x	Type of y	Edge label	Description
Column	Column	SAME-TABLE	x and y belong to the same table.
		FOREIGN-KEY-COL-F	x is a foreign key for y.
		FOREIGN-KEY-COL-R	y is a foreign key for x.
Column	Table	PRIMARY-KEY-F	x is the primary key of y.
		BELONGS-TO-F	x is a column of y (but not the primary key).
Table	Column	PRIMARY-KEY-R	y is the primary key of x.
		BELONGS-TO-R	y is a column of x (but not the primary key).
Table	Table	FOREIGN-KEY-TAB-F	Table x has a foreign key column in y.
		FOREIGN-KEY-TAB-R	Same as above, but x and y are reversed.
		FOREIGN-KEY-TAB-B	x and y have foreign keys in both directions.

Figure 3.2: Pre-existing relations in schema

(2) Encoding : relation-aware input encoder and schema linking

Firstly, in relation-aware input encoder component, each node in the Graph G is vectorized to obtain the initial input X. Then superimpose n RAT Layers on top of X to perform relation-aware self-attention. At this time, the relationship between each node will be r_{ij} is also used as training input.

Currently, RATSQL usually uses some techniques for this problem:

- Glove processed through BiLSTM
- Bert pre-trained embedding. Before continuing to the RAT layers, we input X to the BERT and use the most recent hidden states as starting representations.
- In addition, RAT also incorporates other pre-train techniques such as T5, BART, and RoBERTa.

And an important issue is that these techniques both ensure that the initial representations are independent of the relational information.

Secondly, the Schema linking component includes:

- Name-based linking is based on two kinds of occurrences of column or table describes in the question by using the n-gram technique
  - Exact match includes two kinds 1) the gram in question occur exactly in the column describes (COLUMN-QUESTION-EXACT-MATCH) 2)the

gram in question occur exactly in the table describes (TABLE-QUESTION-EXACT-MATCH)

  - Partial match includes two kind 1) the gram in question occur partly in column describes(COLUMN-QUESTION-PARTIAL-MATCH) 2)the gram in question occur partly in table describes (TABLE-QUESTION-PARTIAL-MATCH)

- Value-based linking: Question-schema alignment also occurs when utterance's words or value that appear in the database.

  - Cell Value match is type number, date time (column-question match is number or date)
  - Cell Value match is a word (column-question match is CELLMATCH). We execute a SQL query to do this.

- In memory-Schema Aligment Matrix: Relation-awware attention [15] as a pointer mechanism between every memory element and all the columns/tables to calculate the alignment matrices L^{col}, L^{tab}

$$L_{i,j}^{(col)} = softmax\{\frac{y_i W_Q^{(col)}(c_j^{final} W(col)_K + r_{ij}^K)^T}{\sqrt{d_x}}\}$$
$$L_{i,j}^{(tab)} = softmax\{\frac{y_i W_Q^{(tab)}(t_j^{final} W(tab)_K + r_{ij}^K)^T}{\sqrt{d_x}}\}$$

(3) Decoding

RATSQL based on an abstract syntax tree that Yin and neubig (2017) proposed in [21]. It genarates the SQL as an abstract syntax tree and use depth-first search algorithm for traversing, by using an LSTM to output sequence of decoder actions that either:

- When creating a leaf node or expanding the most recent created node into a grammar rule that show in the appendix A, use APPLYRULE.

$$Pr(a_t = APPLYRULE[R]|a_t, y) = softmax_R(g(h_t))$$
$$f_{LSTM}([a_{t-1}||Z_t||h_{p_t}||a_{p_t}||n_{f_t}], m_{t-1}, h_{t-1})$$

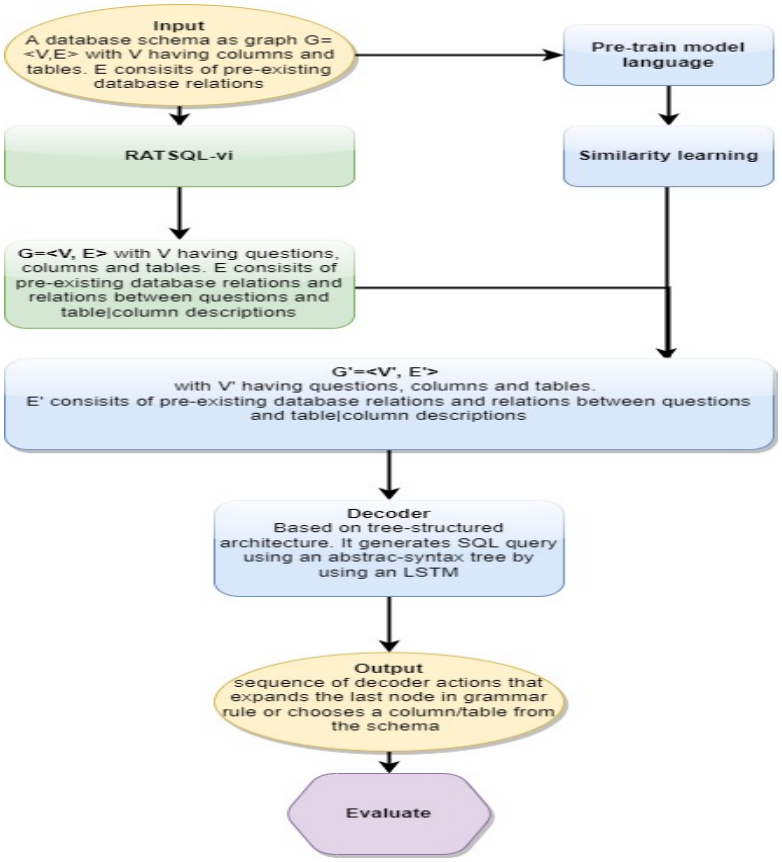
- Select a column/table description from the schema, SELECTCOLUMN, and SELECTABLE

$$\widetilde{\lambda} = \frac{h_t W_Q^{sc} (y_i W_K^{sc})^T}{\sqrt{d_x}}$$
$$\lambda_i = softmax_i(\widetilde{\lambda}_i)$$
$$Pr(a_t = SELECTCOLUMN[i] | a_{<t}, y) = \sum_{j=1}^{|y|} \lambda_j L_{j,i}^{col}$$

Chapter 4

Experimentation

4.1. Implementation diagram



4.2. Configuration and experimentation

Figure 4.1 illustrates the steps in my experiment on Vietnamese Spiders in more detail, from input to evaluation.

(1) Pre-train in my experiments, I used multilingual Bert base and PhoBert base for the pre-train task.

- BERT-Base, Multilingual Uncased : 102 languages include Vietnamese language, 12-layer, 768-hidden, 12-heads, 110M parameters, max length 512
- PhoBERT-Base: Vietnamese language, 150M parameters, max length 256
- BARTPho-Base: Vietnamese language, max length 1024. However, the results will be published shortly due to the long training time.

(2) RatSQL-vi In schema linking component, I assumed that a database's column and table name descriptions were limited to three to seven words. So in Schema linking, I set n-gram to seven. In addition, perform data preprocessing for input questions such as stop word processing and word normalization.

(3) Database translate Current Vietnamese text-to-SQL approaches such as EditSQL or IRnet are not interested in exploiting the relationship between data values of relational databases, so I continued to translate 166 databases of the Spider dataset. However, song titles, music albums or book titles, etc, I still keep the original English.

(4) I use the tokenization that is displayed in the dataset for RATSQL BERT. The dimension of every word embedding is 300. On top of the bidirectional LSTMs, following the original version, I layer 8 relationally aware self-attention layers and I used dropout with a rate of 0.1, dx = dz = 256, H= 8, and these settings within them. In case RAT-SQL and BERT pre-train, I set up a learning rate is 2e-5 to fine-tune BERT with a batch size of 6 and train for up to 70,000 steps.

(5) Decoder Before I put the results into the evaluation, I calibrate the results obtained from my model to the correct AST structure with the table and column names in English and do the same thing with the resulting query in the train, test set.

Chapter 5

Evaluation

5.1. Results

My report results using the same metrics as Yu et al. 2018 [23], however, because I apply a function to map table and column names to their descriptions , so in evaluate part, I also applied a function to map the descriptions to table and column names.

In my experiment, I use the Dataset Spider-vi in the split shown in Table 5.1. Both dev set and train set is splitted to four hardness levels (easy, medium, hard and extra hard) to conduct testing, training and evaluation.

	Quesion	number SQL	number database	Easy	Medium	Hard	Extra hard
all	14616	4082	124	1808	2660	1693	1624
train	6831	3493	99	1559	2255	1502	1515
dev	954	589	25	249	405	191	109

Table 5.1: Spider train set and Spider dev set

The results obtained from the experiment with RATSQL-vi + PhoBert base on the Vietnamese Spider dataset show that the current results are the highest. While RATSQL-vi, when enhanced with mBert, gives lower results due to the limitation of the dictionary of each embedding. However, in both cases, the results show that RATSQL has overcome some errors that previous methods have announced, such

as predicting max and min in the query.

In addition, a problem in the Spider dataset where some databases have a large number of tables and columns thereby increasing the input of the pre-train in both the English and Vietnamese versions of RATSQL ignore this problem and treat in train set.

Approach	Exact
EditSQL [12] [17]	33.7%
EditSQL(Dep) [12] [17]	45.3%
RATSQL + mBERT base without Cell Value match	47.4%
IRNET(DeP) [12] [17]	52.2%
EditSQL (PhoBERT large) [12] [17]	56.7%
IRNET (PHOBERT large) [12] [17]	60.2%
RATSQL + PhoBERT base	**64.7%**

Table 5.2: Results on Spider dev set on vi-Spider dataset

More details on the results obtained by difficulty level are shown in the table 5.3. (RATSQL+BERT v1 is RATSQL + multiligual BERT base without Cell Value match.)

Method	Easy	Medium	Hard	Extra hard	SELECT	WHERE	ORDER BY	GROUP BY	KEYWORD
RATSQL PHOBERT base	79.6	64.9	58.6	40.4	82.5	81.9	78.4	56.8	88.1
RATSQL mBERT v1	61.4	48.2	41.4	22.9	70.4	58.4	72.2	46.9	83.5

Table 5.3: Results on 4 hardness levels sets of dev set on vi-Spider dataset , and F_1 score of SQL components.

5.2. Result Analysis

Since the VinAI team did not publish the source code and results related to this section, I only evaluated and compared them with the results of the VinAi group published in the paper [12].

IRNET PHOBERT method: 18% wrong in predicting operators. For example, given the pharses "già nhất"(oldest) and "trẻ nhất" (youngest) in the utterance,

the method fails to predict the corect operator max and min in [[12],4.3]. However, RATSQL method is more efficiency :

- Question: Độ tuổi trung_bình của các ca_sĩ người Pháp là bao_nhiêu ? Đồng_thời , độ tuổi của ca_sĩ già nhất và độ tuổi của ca_sĩ trẻ nhất đến từ quốc_gia này là bao_nhiêu ?
- Question: What is the average, minimum, and maximum age for all French singers?
  - RATSQL+BERT predict:
    "SELECT Avg(ca_sĩ.tuổi), Max(ca_sĩ.tuổi), Min(ca_sĩ.tuổi) FROM ca_sĩ WHERE ca_sĩ.quốc_gia = 'terminal'"
  - RATSQL+PhoBERT predict:
    "SELECT Avg(ca_sĩ.tuổi), Max(ca_sĩ.tuổi), Min(ca_sĩ.tuổi) FROM ca_sĩ WHERE
    ca_sĩ.quốc_gia = 'terminal'"
  - Gold: "select avg (tuổi) , min (tuổi) , max (tuổi) from ca_sĩ where quốc_gia = "France""
- Question: Hiển_thị tên và quốc_tịch của người tổ_chức bữa tiệc già nhất
- Question: "What is the first name and country code of the oldest player?"
  - RATSQL+mBERT predict:
    "SELECT người_tổ_chức.tên, người_tổ_chức.quốc_tịch FROM người_tổ_chức ORDER BY người_tổ_chức.tuổi Desc LIMIT 1
  - RATSQL+PhoBERT predict:
    "SELECT người_tổ_chức.tên, người_tổ_chức.quốc_tịch FROM người_tổ_chức JOIN chủ_tiệ ON người_tổ_chức.id_người_tổ_chức chủ_tiệc.id_người_tổ_chức ORDER BY người_tổ_chức.tuổi Desc LIMIT 1"
  - Gold: "select tên , quốc_tịch from người tổ_chức order by tuổi desc limit 1"

IRNET PHOBERT method: 32% makes erroneous assumptions about column names that aren't fully or adequately addressed in the questions, [[12],4.3]. In this case, RATSQL predicted correctly, mostly:

- Question: Hiển_thị tên và năm phát_hành của những bài hát thuộc về ca_sĩ trẻ tuổi nhất.
- Question: What are the names and release years for all the songs of the youngest singer?
  - RATSQL+PhoBERT predict: SELECT ca_sĩ.tên_bài_hát, ca_sĩ.năm_phát_hành_bài_hát FROM ca_sĩ ORDER BY ca_sĩ.tuổi Asc LIMIT 1
  - Gold: select tên bài hát , năm phát_hành bài hát from ca_sĩ order by tuổi limit 1

Chapter 6

Conclusion and future work

6.1. Conclusion

When researching some solutions for the text-to-sql problem, I find out and selected solutions that can solve the text-to-SQL problem for the Vietnamese language. The RATSQL method brings high efficiency, especially on the Spider dataset in the Vietnamese language has proven to be far more accurate than previous methods of VinAI team. However, this method also has an inconvenience for practical application implementation. RATSQL uses many SQL queries in Schema linking, which slows down the processing compared to other methods.

6.2. Future work

To apply the RATSQL framework effectively in a short period from 2021-2022, many researchers have published and proposed several methods to improve, perfect and exploit in their work. For example, the SMBOP+GRAPPA method changes the decoder part from a top-down tree to a bottom-up tree, increasing the application of grammar processing in the encoder. According to the team of authors, it has helped 2.2x speed up decoding time.

However, most of these methods are only effective for the English language. Therefore, shortly, I will find solutions to increase the efficiency of processing problems in Vietnamese. Specifically, we will find a solution to deal with Viet-

namese grammar in the encoder and handle synonyms or enhance the handling of the r relationship between the utterance and the table and column names in the database by turning the no-weighted graph to using directed and weighted graphs.

In my opinion, each word and relationship r in the RATSQL graph is represented by a uniform vector of dimensions. However, I find that between different dimensions, there will be a difference between the relationship r, so changing from representing an unweighted relationship r to a weighted relationship W.r and training with a deep learning model will be able to provide an effective research direction.

Acknowledgements

Thank you teacher - professor Nguyen Le Minh has taught me thoroughly and in detail so that I have enough knowledge and apply them to this essay. I thank all the members of Nguyen's lab for the discussions and seminars that helped me find the text to SQL problem-solving approach. Thank you for the enthusiastic guidance of the professor to help me complete my thesis. Also, I would like to thank Jaist's computer center for ensuring the GPU server system helped me achieve my work.

Due to my limited knowledge, there are still many shortcomings and limitations. I hope for the guidance and contributions of the teachers to make my thesis more complete. Sincerely thank!

References

[1] Section most popular technologies databases. https://survey.stackoverflow.co/2022/#section-most-popular-technologies-databases, 2022 Accesssed: 2022-5.

[2] Cao, R., Chen, L., Chen, Z., Zhao, Y., Zhu, S., and Yu, K. LGESQL: line graph enhanced text-to-sql model with mixed local and non-local relations. *CoRR abs/2106.01093* (2021).

[3] Clark, K., Luong, M., Le, Q. V., and Manning, C. D. ELECTRA: pre-training text encoders as discriminators rather than generators. *CoRR abs/2003.10555* (2020).

[4] Devlin, J., Chang, M., Lee, K., and Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR abs/1810.04805* (2018).

[5] Dinh, Q., Phuong, L.-H., Nguyen, H., Nguyen, C.-T., Rossignol, M., and Vu, X. Word segmentation of vietnamese texts: a comparison of approaches.

[6] Guo, J., Zhan, Z., Gao, Y., Xiao, Y., Lou, J., Liu, T., and Zhang, D. Towards complex text-to-sql in cross-domain database with intermediate representation. *CoRR abs/1905.08205* (2019).

[7] Kumar, A., Nagarkar, P., Nalhe, P., and Vijayakumar, S. Deep learning driven natural languages text to sql query conversion: A survey, 2022.

[8] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. BART: denoising sequence-to-sequence

pre-training for natural language generation, translation, and comprehension. *CoRR abs/1910.13461* (2019).

[9] Lin, X. V., Socher, R., and Xiong, C. Bridging textual and tabular data for cross-domain text-to-SQL semantic parsing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, Association for Computational Linguistics (Online, Nov. 2020), 4870–4888.

[10] Liu, A., Hu, X., Lin, L., and Wen, L. Semantic enhanced text-to-sql parsing via iteratively learning schema linking graph, 2022.

[11] Luong Tran, N., Le, D. M., and Nguyen, D. Q. BARTpho: Pre-trained Sequence-to-Sequence Models for Vietnamese. *arXiv e-prints* (Sept. 2021), arXiv:2109.09701.

[12] Nguyen, A. T., Dao, M. H., and Nguyen, D. Q. A pilot study of text-to-sql semantic parsing for vietnamese. *CoRR abs/2010.01891* (2020).

[13] Pires, T., Schlinger, E., and Garrette, D. How multilingual is multilingual bert? *CoRR abs/1906.01502* (2019).

[14] Rubin, O., and Berant, J. Smbop: Semi-autoregressive bottom-up semantic parsing. *CoRR abs/2010.12412* (2020).

[15] Shaw, P., Uszkoreit, J., and Vaswani, A. Self-attention with relative position representations. *CoRR abs/1803.02155* (2018).

[16] Shi, P., Ng, P., Wang, Z., Zhu, H., Li, A. H., Wang, J., Santos, C. N. d., and Xiang, B. Learning contextual representations for semantic parsing with generation-augmented pre-training, 2020.

[17] Tuan Nguyen, A., Dao, M. H., and Nguyen, D. Q. A pilot study of text-to-SQL semantic parsing for Vietnamese. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, Association for Computational Linguistics (Online, Nov. 2020), 4079–4085.

[18] Wang, B., Lapata, M., and Titov, I. Meta-learning for domain generalization in semantic parsing. *CoRR abs/2010.11988* (2020).

[19] Wang, B., Shin, R., Liu, X., Polozov, O., and Richardson, M. RAT-SQL: relation-aware schema encoding and linking for text-to-sql parsers. *CoRR abs/1911.04942* (2019).

[20] Wang, B., Shin, R., Liu, X., Polozov, O., and Richardson, M. 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*, Association for Computational Linguistics (Online, July 2020), 7567–7578.

[21] Yin, P., and Neubig, G. A syntactic neural model for general-purpose code generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics (Vancouver, Canada, July 2017), 440–450.

[22] Yu, T., Wu, C., Lin, X. V., Wang, B., Tan, Y. C., Yang, X., Radev, D. R., Socher, R., and Xiong, C. Grappa: Grammar-augmented pre-training for table semantic parsing. *CoRR abs/2009.13845* (2020).

[23] Yu, T., Yasunaga, M., Yang, K., Zhang, R., Wang, D., Li, Z., and Radev, D. R. Syntaxsqlnet: Syntax tree networks for complex and cross-domaintext-to-sql task. *CoRR abs/1810.05237* (2018).

[24] Zhang, R., Yu, T., Er, H. Y., Shim, S., Xue, E., Lin, X. V., Shi, T., Xiong, C., Socher, R., and Radev, D. Editing-based sql query generation for cross-domain context-dependent questions, 2019.

Appendix

A. Grammar

Assumptions:

1. sql is correct
2. only table name has alias
3. only one intersect/union/except

module Spider

$\{val = Number(objectf)|String(strings)$

$|ValSql(sqls)|ColUnit(col\_unitc)|Terminal$

$col\_unit = (agg\_typeagg\_id, columncol\_id, singletonis\_distinct)$

$val\_unit = Column(col\_unitcol\_unit1)|Minus(col\_unitcol\_unit1, col\_unitcol\_unit2)$

$|Plus(col\_unitcol\_unit1, col\_unitcol\_unit2)|Times(col\_unitcol\_unit1, col\_unitcol\_unit2)$

$|Divide(col\_unitcol\_unit1, col\_unitcol\_unit2)$

$table\_unit = TableUnitSql(sqls)|Table(tabletable\_id)$

$cond = And(condleft, condright)|Or(condleft, condright)|Not(condc)$

$|Between(val\_unitval\_unit, valval1, valval2)$

$|Eq(val\_unitval\_unit, valval1)|Gt(val\_unitval\_unit, valval1)|Lt(val\_unitval\_unit, valval1)$

$|Ge(val\_unitval\_unit, valval1)|Le(val\_unitval\_unit, valval1)|Ne(val\_unitval\_unit, valval1)$

$|In(val\_unitval\_unit, valval1)|Like(val\_unitval\_unit, valval1)$

$sql = (selectselect, fromfrom, sql\_wheresql\_where,$

$sql\_groupbysql\_groupby, sql\_orderbysql\_orderby,$

$sql\_ieusql\_ieu,)$

$sql\_where = (cond?where,)$

$sql\_groupby = (col\_unit * group\_by, cond?having,)$

$sql\_orderby = (order\_by?order\_by, int?limit,)$

$sql\_ieu = (sql?intersect, sql?except, sql?union,)$
$select = (singletonis\_distinct, agg * aggs)$
$agg = (agg\_typeagg\_id, val\_unitval\_unit)$
$from = (table\_unit * table\_units, cond?conds)$
$order\_by = (orderorder, val\_unit * val\_units)$
$agg\_type = NoneAggOp|Max|Min|Count|Sum|Avg$
$order = Asc|Desc\}$