分类号_____ 学号 M201973306
学校代码 10487 密级_____

华中科技大学

硕士学位论文

基于深度学习的中文自然语言生成复杂 SQL 语句技术研究

学位申请人：林毅炜

学 科 专 业：计算机技术

指 导 教 师：辜希武　副研究员

答 辩 日 期：2021 年 5 月 23 日

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Professional Master Degree

Research on Generating Complex SQL Statements from Chinese Natural Language Based on Deep Learning

Candidate : LIN Yiwei

Major : Computer Technology

Supervisor : Assoc Prof. GU Xiwu

Huazhong University of Science and Technology

Wuhan 430074, P. R. China

May, 2021

摘 要

在信息技术的高速发展的现代社会，海量的数据常结构化存储于数据库中，在检索数据时，需要用到统一数据库查询语言 SQL。但 SQL 作为有严格语法规则约束的结构化查询语言，需要用户具备数据库和 SQL 专业知识，使用门槛较高，对非专业用户不友好。近年来，为了提高数据库的信息检索效率、降低用户的使用门槛，使数据库能更好服务于大众，由计算机将自然语言问题直接生成为 SQL 语句的研究得到了人们的关注，该研究任务被称为 Text-to-SQL 任务。

Text-to-SQL 作为一项综合性任务，需要解决自然语言编码、数据库模式编码、关系发现、关系编码、SQL 语句解码等多个关键问题。目前已有的研究工作大多建立在自然语言问题与目标数据库模式均为英文的基础上，并且对 SQL 语句的结构做了大量简化，无法对复杂的 SQL 语句进行处理，存在诸多局限性，难以满足工程实践的需要。

针对中文复杂 Text-to-SQL 任务的特点和目前研究存在的问题，提出了中文自然语言生成复杂 SQL 的模型 RACN-SQL，通过语义编码、构建数据库模式关系表示、模式链接、关系编码、SQL 解码，实现从自然语言问题到 SQL 语句的转换，最终输出 SQL 查询结果。

与其他研究相比，RACN-SQL 模型的主要贡献点在于：（1）解决了中文自然语言问题与英文数据库模式之间的跨语言语义编码障碍；（2）合理、有效地将自然语言与数据库模式之间的关联关系发现并表示出来；（3）综合考虑非结构化数据和结构化数据之间的特点，将语义信息和关系信息结合起来联合编码。

实验中选取符合实际需求的中文复杂 Text-to-SQL 数据集和评价指标，通过与其他经典模型和优秀模型对比实验，验证提出的 RACN-SQL 模型的有效性，同时分析模型存在的不足和未来研究需要关注的方向。

关键词： 自然语言生成 SQL；信息检索；语义编码；模式链接；关系编码

Abstract

Due to the development of information technology is really fast, massive data are stored in the database structurally. When retrieving the data, it is necessary to use the unified database query language SQL. However, as a structured query language with strict grammar rules, SQL requires users to have professional knowledge of database and SQL, which has a high threshold for use and is not friendly to non-professional users. In recent years, in order to improve the efficiency of database information retrieval and reduce the user's threshold to make the database better serve public, the research on the direct generation of natural language problems into SQL statements by computers has attracted people's attention. This research task is also known as Text-to-SQL task.

As a comprehensive task, Text-to-SQL needs to solve many key problems, such as natural language encoding, database schema encoding, relational discovery, relational encoding, SQL statement decoding and so on. At present, most of the existing research work is based on the natural language problem and the target database schema is in English, and the structure of SQL statements has been greatly simplified, so it is impossible to deal with the complex SQL statements, and there are many limitations, it is difficult to meet the needs of engineering practice.

Aiming at the characteristics of complex Chinese Text-to-SQL tasks, the RACN-SQL model to generate complex SQL from Chinese natural language is proposed after analyzing the existing problems in the current research. Through semantic encoding, database schema relational representation, schema linking, relational encoding, and SQL decoding, the model realizes the conversion from natural language questions to SQL statements, and finally output SQL query results.

Compared with other researches, the main contributions of the RACN-SQL model are as follows: (1) solving the cross-language semantic encoding barrier between the Chinese natural language problem and the English database schema; (2) finding and expressing the correlation between natural language and database schema reasonably and effectively; (3) Synthetically considering the characteristics between unstructured data and

structured data, the semantic information and relational information are combined for joint encoding.

In the experiment part, complex Chinese Text-to-SQL dataset and evaluation methods that meet the actual needs are selected. Through comparative experiments with other classic models and excellent models, the effectiveness of the proposed RACN-SQL model is verified. Meanwhile, the existing deficiencies of the model and the direction that future research needs to pay attention to are analyzed.

Key words: Text-to-SQL, Information Retrieval, Semantic Encoding, Schema Linking, Relational Encoding

目 录

1 绪论

1.1 课题研究背景

随着数据库技术、信息技术的高速发展，在各行各业，每时每刻都在产生海量的数字化数据，数据之间可能是彼此独立的，也可能存在一定的依赖关系。因此为了方便数据的查询和更新、统一进行管理和维护，这些数据通常结构化存储于数据库中，从数据库中检索符合特定要求的数据就需要用到统一数据库查询语言 SQL。

结构化的数据库查询语言 SQL 有严格的语法约束，对于不具备数据库和 SQL 相关知识的非专业用户，有一定的学习和使用门槛。随着近年来深度学习技术和自然语言技术发展，通过自然语言直接进行数据库查询以实现更便捷、高效的检索信息方式得到了人们的广泛关注。

自然语言具有语法较为灵活、用词和表达方式随意、多样等特点，而 SQL 作为一门结构化查询语言，有着较为严格的语法，因此需要机器“理解”自然语言问题要表达的语义后，根据 SQL 的语法规则生成对应的 SQL 语句。传统的基于句法语法分析的自然语言生成 SQL 方法通过对自然语言进行分析，构建大量语法解析的模板，由语法树来生成最终的 SQL 语句，这种方法对自然语言的表达有严格要求，难以解决口语化、缺失成分等问题，并且生成的 SQL 结构简单，不能满足生产生活中多样的需求。通过深度学习技术和自然语言技术，将用户提出的自然语言问题的中的词逐一转换成向量，经过训练后，相近含义的词拥有近似的向量，能解决自然语言表达多样性的问题。同时，经过训练的模型能够根据问句和数据表的信息，从问句中提取有价值的特征，不需要领域内专家来制定繁杂的规则，降低人工的维护成本。

在语义解析领域，将自然语言表达的问句生成对应 SQL 语句的任务称为 Text-to-SQL 任务。相较于目前广泛开展的英文 Text-to-SQL 任务研究而言，中文 Text-to-SQL 的研究则更为复杂，现有的研究工作少，并且这些研究大多着眼于简单的 SQL 语句。中文 Text-to-SQL 任务的难点在于中文的问句中不存在英文中的分词符，因此产生歧义的几率高，加之中文表述上的同义词更多、表达方式更加多样，

进一步增加了机器理解的难度。在生产实践中，存储于数据库的数据表名、数据列名等字段通常以英文存储，这更增加了从中文自然语言表达推断其所指代的数据表和数据列的难度。

1.2 课题研究目的和意义

随着现代社会信息化程度越来越高，存储在数据库中的数据量呈指数级增长，要从中找到符合条件的数据较为繁琐。在实际应用场景中，不具备数据库和 SQL 专业知识的用户也有大量信息检索的需求，为了降低用户的学习和使用成本，需要计算机能够“理解”用户的检索需求，并生成 SQL 查询语句交由数据库执行引擎执行，返回最终的结果。即实现自然语言查询接口（Natural Language Interface，NLI），用户通过该接口直接与数据库交互进行信息检索，来达到提高效率和降低使用门槛的目的。

Text-to-SQL 在商业上也有广泛需求和研究价值，如语音助手、对话系统等。将用户提出的问题生成 SQL 语句后到后台数据库中进行检索，能快速回复用户答案，目前广泛应用的场景有智能客服、企业表格数据的快速检索、个性化问题的解答等，这种自动化的问答方式能够降低大量的人力成本。

综合目前 Text-to-SQL 领域的研究工作，自 2017 年起，陆续有基于深度学习方法的英文 Text-to-SQL 研究，但中文相关研究工作甚少。同时在现有的研究工作中，通过简化 SQL 语法结构、忽略数据表定位等方式简化模型结构，虽然在一定程度上提高了准确率，但不具备工程上的实用性。比如针对 WikiSQL 数据集开展的一系列研究中，SQL 语句只有 SELECT、FROM、WHERE 关键字，这种简单的 SQL 查询场景不能满足复杂的应用需求。目前工程实践中需求的中文 Text-to-SQL 需要支持所有结构的 SQL 语句生成，同时中文自然语言问题与英文数据库模式之间存在跨语言的特性，数据库还具有更新迭代快、安全隐私要求高等特点，目前的研究工作难以满足工程实践的需要。

本文旨在基于深度学习技术，研究中文复杂 Text-to-SQL 任务的模型和方法，具有重要的理论意义和实际应用价值。

1.3 国内外研究现状

1.3.1 中文自然语言编码研究

自然语言编码是 Text-to-SQL 任务中语义编码的重要一环，它将用户提问的自然语言转换为机器能够识别的数据结构并从中提取特征。自然语言存在的形式通常是一段文本，在计算机中体现为一段字符串。对于英文文本，各单词之间有空格作为分隔符，但中文文本的词之间没有分隔符，计算机难以理解一长串没有分隔符的文本。因此中文语料在进行编码之前还需进行分词操作，目前准确率较高的常用中文分词工具有 Jieba[1]、哈工大语言技术平台[2]等。

分词后，再对中文文本进行词嵌入（Word Embedding），即用向量来表示句子中的每个词，词向量常用独热（one-hot）编码、随机初始化、或通过 GloVe[1]、Word2Vec[2]等词向量模型训练得到，然后再用循环神经网络[3]（Recurrent Neural Network，RNN）或自然语言的预训练模型[4]进行编码。

循环神经网络与卷积神经网络[5]（Convolutional Neural Network，CNN）相比，更适用于序列化数据，有更好的特征捕捉能力，符合自然语言处理的特点，广泛应用于文本编码任务中。但传统的循环神经网络对于文本中距离较长的依赖关系，捕捉效果欠佳，而且对于长文本的训练，它还存在着梯度消失和梯度爆炸的问题，因此目前已有许多工作对它进行改进。比如长短期记忆网络[6]（Long Short-Term Memory Network，LSTM）通过三个门（输入门、输出门、遗忘门）来控制每个时刻中序列信息的传递，相较于循环神经网络，LSTM 对序列中的长期依赖关系有更强的捕捉能力。另外，也有一些工作[7][8]对长短期记忆网络进行了改进，采用双向 LSTM（Bi-LSTM）进行文本编码，从而解决文本的上下文关联问题。

目前广泛使用预训练模型来完成自然语言处理的下游任务主要有两种策略，第一种是以 ELMo[9]为代表的基于特征的方法，使用特定领域的文本特征，提取出上下文敏感的特征；另一种是以 Open AI GPT[10]为代表的基于微调的方法，这种方法会在

[1] https://pypi.org/project/Jieba/

[2] https://www.ltp-cloud.com/

已训练好的预训练模型基础上，微调可训练参数以适应目标任务，训练速度更快，学习的成本更低。2018 年由 Google 公司发布的预训练模型 BERT[11]（Bidirectional Encoder Representation from Transformers），在 Transformer 模型的编码（Encoder）部分基础上，利用注意力（Attention）机制[12]，同时使用大量语料进行训练，在下游任务的效果上，较之前的模型都有显著提升。

1.3.2 传统的自然语言生成 SQL 研究

Text-to-SQL 任务在近几十年来都是数据库领域的研究热点，早期传统的研究[13]大多是通过对特定数据库人工制定匹配规则的方式来完成。

在英文领域，用于月球岩石分析的自然语言查询系统 Lunar[14]、与用户进行对话交互以消除歧义来提高查询精确度的 Rendezvous[15]、用于大型数据库的查询系统 Ladder[16]、基于 Prolog 表达式实现的 Chat-80[17]等是英文 Text-to-SQL 任务的典型代表。

在中文领域，基于语法和语义结构的 EAAD 模型[18]、将自然语言转为语义依存树进而生成 SQL 语句的 Nchiql[19]系统以及使用 WordNet[20]将自然语言进行语法标注并将词映射到词典中进而生成 SQL 的 PRECISE[21]系统等是中文 Text-to-SQL 任务的典型代表。

1.3.3 基于深度学习的自然语言生成 SQL 研究

基于深度学习的 Text-to-SQL 研究集中在近五年。

2016 年，Li 等人用编码-解码的思想[22]来解决 Text-to-SQL 任务，提出了两种深度学习模型，一是简单的利用 Seq2Seq[23]将 Text-to-SQL 任务视为序列生成任务处理；二是通过 Seq2Tree 将解码器改为分层树的结构，用来捕捉 SQL 语句中的结构信息。

2017 年，Victor 等人提出了 Seq2SQL[24]以及 WikiSQL[3]英文数据集，图 1-1 为该数据集中的一条数据，该模型借鉴了 Seq2Seq 思想，使用插槽思想固定 SQL 语句中的关键字：SELECT、FROM、WHERE，将预测 SQL 语句的任务转换为若干个子任务：预测 SELECT 子句中的聚合操作符、预测 SELECT 子句的列、预测 WHERE 子

[3] https://github.com/salesforce/WikiSQL

句的约束条件等，将输入的问题进行编码（encode），然后解码（decode）为对应的SQL 语句输出，此外，该模型应用强化学习[25][26]来传统 Seq2Seq 模型解决 WHERE 子句中多个约束条件的顺序性问题，在 WikiSQL 上分别达到逻辑正确率 48.3%和执行正确率 59.4%。

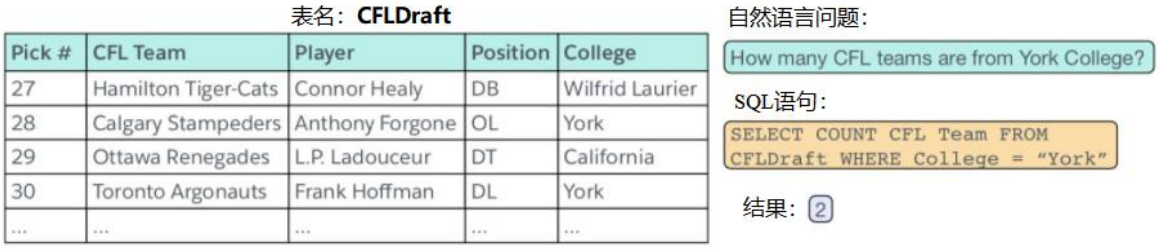

图 1-1 WikiSQL 数据集中的一条数据示例

Xu 等人对 Seq2SQL 进行了改进，提出了 SQLNet[27]。他们认为 Seq2SQL 中的强化学习并不能很好的解决 WHERE 子句中多个约束条件的顺序性问题，因此使用序列到集合（Seq2Set）结构和列注意力机制取代了 Seq2SQL 中使用强化学习预测 WHERE 子句的约束条件，最终在 WikiSQL 数据集上获得了 9%-13%的准确率提高。

Yu 等人提出的 TypeSQL[28]也应用了 Seq2Set 结构，通过外部知识库对问题中的实体类型和表格中的字段信息进行编码，同时考虑了不同子任务之间的依赖关系，取得了更好的效果。

Shi 等人提出的 IncSQL[29]则应用了序列到行为（Seq2Act）的结构，该结构仍基于插槽填充的思想，考虑具有相同或相近语义的 SQL 语句来训练模型，将预定义的行为插入到槽中，生成最终的 SQL 语句。

去年以来，随着以 BERT[11]为代表的预训练语言模型在自然语言处理任务上取得良好的成绩，研究者们也开始将预训练语言模型应用到 Text-to-SQL 任务上。比如去年由 Wonseok 等人提出的 SQLova[30]使用 BERT 作为模型的输入表示层，以此取代 SQLNet 中的词向量表示。SQLova 将预测 SQL 语句任务划分为 6 个子任务，并将自然语言提出的问题和表格中各列的列名均作为网络的输入进行编码。

监督学习模型通过大量标注数据进行学习能保证较高的准确率，但不可否认的是数据标注的成本是极高的，特别是对于 Text-to-SQL 任务，需要人工针对每个问题

和对应数据库表的结果撰写 SQL 语句。也正是由于标注成本高，目前的监督学习模型大多都是在 WikiSQL 公开数据集上进行实验与比较，除此之外，其他数据集大多由于数据量少、数据覆盖领域有限等原因，实验结果的说服力不高。因此，研究者也开始关注弱监督学习[31]在 Text-to-SQL 任务上的应用。

在弱监督学习模型中，以执行正确率作为评价指标，即以自然语言转换的 SQL 语句的执行结果作为更新模型参数的依据。Chen 等人提出的 MAPO[32]（Memory Augmented Policy Optimization）将 Text-to-SQL 任务作为强化学习任务，把输入的自然语言问题和数据库模式作为 Text-to-SQL 强化学习的状态，把当前状态下可能产生的程序集合作为动作空间，预测过程中，MAPO 会根据当前环境生成策略函数，该策略函数以输入的自然语言问题来计算每个可能程序的概率分布，最后选取概率值最高的作为 SQL 语句的预测结果，MAPO 在 WikiSQL 数据集上达到了 74.9%执行正确率，超过了很多监督学习的模型。但也有研究者注意到 MAPO 中会出现虚假回报和稀疏回报的问题，这降低了模型的准确率，因此 Rishabh 等人提出通过构建辅助奖励来弥补虚假回报问题的 MeRL[33]（Meta Reward Learning），它通过辅助奖励机制对学习中的参数调整，对环境的改变产生精准反馈，另外针对稀疏回报情况下更高效搜索样本空间的问题，提出对 MAPO 的改进方案 MAPOX，经过实验验证，取得了更好的训练效果。

1.3.4 基于深度学习的自然语言生成复杂 SQL 研究

```
SELECT  $AGG  $COL
WHERE   $COL  $OP  $VAL
(AND    $COL  $OP  $VAL)*
```

图 1-2 WikiSQL 数据集槽填充图

虽然 WikiSQL 数据集提供了大量的人工标注样本，但数据集中的 SQL 语句较简单，因此在 WikiSQL 数据集上研究所使用的槽填充模型也较简单，如图 1-2 所示，仅包含 SQL 语句中 SELECT 和 WHERE 关键字，预测任务转化为对$AGG（聚合函数）、$COL（列）、$OP（运算符）、$VAL（值）的预测，“*”表示该部分出现 0 次或以上。

图 1-3 Spider 数据集中的一条数据

然而在实际应用中，SQL 语句往往更为复杂，如表连接 JOIN、分组操作 GROUP BY 与 HAVING、排序操作 ORDER BY，以及嵌套查询等。那么上述针对 WikiSQL 数据集的研究工作并不能很好地满足这些复杂 SQL 生成的需求。因此，Yu 等人的开始关注英文复杂 Text-to-SQL 任务[34]，标注并发布了 Spider 数据集[4]，如图 1-3 为该数据集的一个实例，数据集中有大量复杂的 SQL 语句，比如该实例中就包含了表连接操作 JOIN、分组操作 GROUP BY 和 HAVING、嵌套查询。

随着 Spider 数据集的公开发布，出现了对英文复杂 Text-to-SQL 任务的一系列研究工作。

Ben 等人提出的 GNN 模型[35]针对复杂的数据库模式，用图来表示不同数据表、数据列之间的关系，如主键、外键关联等，通过图结构编码数据库模式后与对自然语言编码后的向量共同输入神经网络，使得网络有能力对表连接、子查询等数据表、数据列的关联操作进行预测。

[4] https://github.com/taoyds/spider

Guo 等人提出的 IRNet 模型[36]将自然语言转换为一种树形中间语言 SemQL，再将 SemQL 转为 SQL 语句。此外，该模型使用 n-gram 将自然语言中指代的数据表、数据列等进行对齐，来精准定位自然语言问题描述的数据表和数据列，整个过程分为自然语言编码、数据库模式编码以及解码三个部分，最终在 Spider 测试集上获得了 46.7%的准确率。

Zhang 等人提出的 EditSQL 模型[37]通过交互的方法来提高生成 SQL 语句的准确率，取得了不错的成绩。模型首先根据用户的提问和数据库模式信息生成一个初始的 SQL 语句，由用户来判断该 SQL 语句中哪些部分不符合查询需求，模型不断进行修改，直到生成正确的 SQL 语句。EditSQL 模型将 SQL 语句视为序列，使用基于 BERT 的问题-表编码器和表感知解码器，在 Token 级别生成关键字。

Choi 等人提出的 RYANSQL（Recursively applying sketch-based slot fillings for complex text-to-sql in cross-domain databases）模型[38]在 Spider 数据集上取得了很好的成绩，该模型通过 SQL 语句的位置码（Statement Position Code，SPC）来标记 SQL 语句的复杂结构的方式，提出了基于草图的槽填充方法。作者首先提出了 SELECT 语句的草图和填充网络结构，这个结构表示出了 SQL 语句的生成语法规则，模型按照填充网络结构就能生成语法正确的 SQL 语句，然后根据 SPC 的信息，表示出 SQL 语句的层次，解决嵌套查询等复杂的场景。另外，RYANSQL 还使用了两个针对模型输入的启发式方法，降低了输入噪声的干扰。

Wang 等人提出的 RAT-SQL（Relation-Aware Schema Encoding and Linking for Text-to-SQL Parsers）模型[39]目前在 Spider 数据集上表现最优，达到了 65.6%准确率。受到 Shaw 等人将自注意力用于关系位置表示[40]的启发，该模型基于自注意力机制进行了改进，通过增加偏置来描述数据库中已知的关系，并通过基于名字和基于值的自然语言问题和数据库模式的关联策略进行编码，最后使用抽象语法树进行解码得到最终的 SQL 语句。

为了进一步提高模型的准确率，Li 等人[41]和 Yao 等人[42]先后提出了基于人机交互不断修正所生成的 SQL 语句的方法，模型在分析自然语言问题时，当对某个词所指代的数据表名、数据列名或值不确定时，会向用户提出问题，帮助模型来确定。

Easy

What is the number of cars with more than 4 cylinders?
有多少辆车的汽缸超过 4 个?

```
SELECT COUNT(*)
FROM cars_data
WHERE cylinders > 4
```

Medium

For each stadium, how many concerts are there?
每个体育馆开过多少演唱会?

```
SELECT T2.name, COUNT(*)
FROM concert AS T1 JOIN stadium AS T2
ON T1.stadium_id = T2.stadium_id
GROUP BY T1.stadium_id
```

Hard

Which countries in Europe have at least 3 car manufacturers?
欧洲有哪些国家至少有 3 家汽车制造商?

```
SELECT T1.country_name
FROM countries AS T1 JOIN continents
AS T2 ON T1.continent = T2.cont_id
JOIN car_makers AS T3 ON
T1.country_id = T3.country
WHERE T2.continent = 'Europe'
GROUP BY T1.country_name
HAVING COUNT(*) > 3
```

Extra Hard

What is the average life expectancy in the countries where English is not the official language?
在那些英语不是官方语言的国家，平均预期寿命是多少?

```
SELECT AVG (life_expectancy)
FROM country
WHERE name NOT IN
    (SELECT T1.name
    FROM country AS T1 JOIN
    country_language AS T2
    ON T1.code = T2. country_code
    WHERE T2.language = "English"
    AND T2.is_official = "T")
```

图 1-4 CSpider 数据集中不同难度数据示例

对于 Text-to-SQL 这种需要以非结构化的自然语言问题来关联结构化的数据库数据的任务而言，序列到序列的预训练模型效果表现并不佳。最近，出现了对 Text-to-SQL 任务的预训练模型研究，Zhao 等人提出的 GP 模型[43]和 Shi 等人提出的 GAP 模型[44]，相较于传统的预训练模型使用的遮盖语言模型（Masked Lanuage Model，MLM）,增加了列预测（Column Prediction）、列恢复（Column Recovery）、SQL 生成（SQL Generation）等新的预训练学习目标，来增强模型中编解码器对结构

化的数据库表和非结构化的自然语言表达之间的理解和推理能力，实验表明，使用这些预训练模型对原模型的表现有一定提高。

2019 年，Min 等人将 Spider 复杂 SQL 数据集中的自然语言问题部分进行了中文翻译[45]，发布了 CSpider 中文复杂 SQL 数据集[5]，为了与工程实践中的做法相统一，作者保留了英文的数据表名和数据列名，因此该数据集相较于 Spider 数据集，在难度上有显著提升。图 1-4 为 CSpider 数据集中不同难度数据的示例，其划分依据是 SQL 中关键字的数量、嵌套查询等。在论文中，作者将 Yu 等人提出的 SyntaxSQLNet 模型[46]进行了修改，作为 CSpider 数据集的 baseline 模型，在测试集上准确率最高为 12.1%。

1.3.5 目前国内外研究的不足

早期的研究是针对特定领域的数据库通过人工制定规则的方法实现，不具备通用性，且需要大量人力维护成本。2017 年以来，围绕 WikiSQL 数据集展开的研究中 SQL 语句形式过于简单，仅以 SELECT、FROM 和 WHERE 构成，难以满足工程实践的需要。2018 年，Spider 复杂 SQL 数据集发布，才陆续有对英文复杂 SQL 生成的研究。2019 年，CSpider 数据集发布，中文复杂 SQL 生成的研究在语义解析领域引起了一定关注。但由于机器更难以理解中文语义，且中文自然语言表达的问句和英文表示的数据表、数据列以及数据值之间有跨语言的障碍，大大增加了中文 Text-to-SQL 的难度，因此目前对中文复杂 SQL 生成的研究甚少。

纵观近年复杂 SQL 生成的研究，使用相对独立的编码方法分别对自然语言和数据库模式编码，很少考虑已有的数据库模式与用户提问之间的联系，这显然忽略了自然语言和数据库模式之间的联系。专业人员在将问题转换为对应 SQL 语句时，必然熟悉数据库中已有的数据表、数据列以及不同列的类型和可能的数据值等信息，在理解用户提问需求的基础上，运用 SQL 语法规则编写 SQL 语句。因此直观上来说，模型应充分捕捉自然语言问题与数据库模式之间的关系，包括直接指代和隐含的常识性关联，并将问句与模式联合编码，有助于进一步提高模型的准确率。

[5] https://github.com/taolusi/chisp

在实际场景中，数据库中的数据表和数据列往往十分复杂，常出现不同表中同名数据列的情况，因此，明确自然语言问题中指代的数据列、数据表、数据值类型都有利于消除歧义，提高准确率，但目前的研究很少考虑。

在最终生成 SQL 语句时，多数研究直接将特征向量解码为 SQL 语句，忽略了复杂 SQL 语句的语法规则，比如同样是条件约束，一般将约束条件放在 WHERE 子句中，但 GROUP BY 之后的条件应放在 HAVING 子句中。因此直接解码过程常因语法约束的复杂性导致准确率降低。

1.4 本文主要研究内容

针对目前研究存在的问题，本文从数据库模式的结构化数据与自然语言问句的非结构化数据的特点出发，改进目前的深度学习的模型，提出端到端的应用于中文语料的复杂 SQL 生成的模型和方法。

本文具体的研究内容如下：

（1）针对中文 Text-to-SQL 任务的特点，设计面向 Text-to-SQL 任务的中文自然语言问题和英文数据库模式的语义编码表示方法。合理有效的语义编码表示方法需要捕捉用户提出的问题与数据库模式之间的对齐关系，包括隐含的指代关系，并充分利用各数据库列的类型信息，这对下游提高任务的准确率至关重要。

（2）构建数据库模式的关系表示。数据库的模式在数据库建立时就已经显式定义，可以表示为图关系，图的顶点是数据表名或数据列名，边关系可以是主键、外键、同属等。下游任务通过数据库模式的关系判断用户的问题是否涉及到表连接、嵌套查询等多表操作，进而生成复杂 SQL。

（3）链接自然语言问题与数据库模式之间的关系。组成 SQL 语句的成分包括 SQL 关键字以及数据表名、数据列名、数据值，为了理解自然语言问题的检索意图，需要模型准确捕捉自然语言问题中对数据库模式的关联关系。自然语言问题中出现的关联关系主要有直接关联和间接关联两大类。直接关联是指在自然语言问题中直接出现数据库中的表名、列名的全称或一部分；间接关联是指自然语言问题中并不直接出现数据库中的表名或列名，但通过常识推理、语义分析等方式完成的关联。

（4）对自然语言问题和数据库模式进行关系编码。基于数据库模式关系及自然语言问题与数据库模式的链接关系，将这些不同的关系进行向量编码，训练模型捕捉输入向量各部分的关系信息，对于非结构化数据的自然语言问题与结构化数据的数据库模式之间联合编码至关重要，在生成 SQL 语句的解码阶段，SQL 语句中不同表的链接关系、嵌套关系等都需要向量中的关系编码信息，高效、鲁棒的关系编码方法是生成正确结构复杂 SQL 语句的关键。

（5）制定合理的 SQL 语句生成策略。自然语言的问题属于非结构化数据，而 SQL 语句属于有特定语法规则的结构化数据。本文所研究的复杂 SQL 语句存在多条件、多表和嵌套等查询，合理的 SQL 语句生成策略有助于解码器根据编码后的向量按照语法规则生成正确的 SQL 语句。

（6）设计实验验证本文提出的中文 Text-to-SQL 方案的有效性。实验部分将选取目前在英文 Text-to-SQL 任务中表现良好的模型，将其应用于中文 Text-to-SQL 任务，选取合适的中文数据集和评价指标进行对比实验。

1.5 论文组织结构安排

本文共分五章节，各章节的组织结构安排如下：

第一章为绪论部分，阐述中文复杂 Text-to-SQL 任务的研究背景、研究目的和意义、国内外研究现状以及本文的主要研究内容。

第二章为相关技术综述部分，详述本文所提出模型流程中所涉及的相关技术，包括理论、结构和用法。

第三章为模型介绍部分，详细介绍本文提出的中文自然语言生成复杂 SQL 的模型 RACN-SQL，将模型的流程细化为 5 个关键步骤：语义编码、数据库模式表示、模式链接、关系编码、SQL 语句解码，分别对这 5 个关键步骤从设计思路、算法流程、计算公式、示例说明等多个维度进行阐述。

第四章为实验与分析部分，首先介绍了实验的基本信息，包括实验软硬件环境、使用的数据集、对比模型、各模型参数设置、评价指标等，然后对实验结果数据进行分析评估，来验证本文所提出的 RACN-SQL 模型在中文复杂 Text-to-SQL 任务上

的有效性。

第五章为总结与展望部分。总结部分对本文中文复杂 Text-to-SQL 的研究工作和所提出模型 RACN-SQL 的主要内容和贡献进行总结，展望部分对模型中可能存在的不足之处进行分析，指出未来工作可以进行改进和深入研究的方向。

2 相关技术综述

中文Text-to-SQL任务的目标是根据现有的数据库模式和用户提出的自然语言问题生成对应的SQL语句。由于Text-to-SQL是一个综合性任务，与之相关开展了一系列工作，涉及多项技术，本章节对本文所涉及的中文Text-to-SQL方案中的技术进行介绍。

2.1 Text-to-SQL技术框架

图2-1为Text-to-SQL技术框架图，系统的输入有用户提出的自然语言问题和已存在的数据库模式，分别对它们进行语义编码，得到语义向量；同时，关系发现与关系表示模块将数据库模式中已定义的关系以及自然语言问题与数据库模式之间的链接关系表示为一个关系有向图。关系编码模块利用关系有向图对语义向量进一步关系编码得到关系向量，关系向量是自然语言问题和数据库模式的字符特征、语义特征和关系特征的高维向量表示，SQL解码模块接受关系向量和数据库模式，根据SQL语法规则生成SQL语句，数据库执行引擎通过执行所生成的SQL语句，得到查询结果作为输出返回给用户。

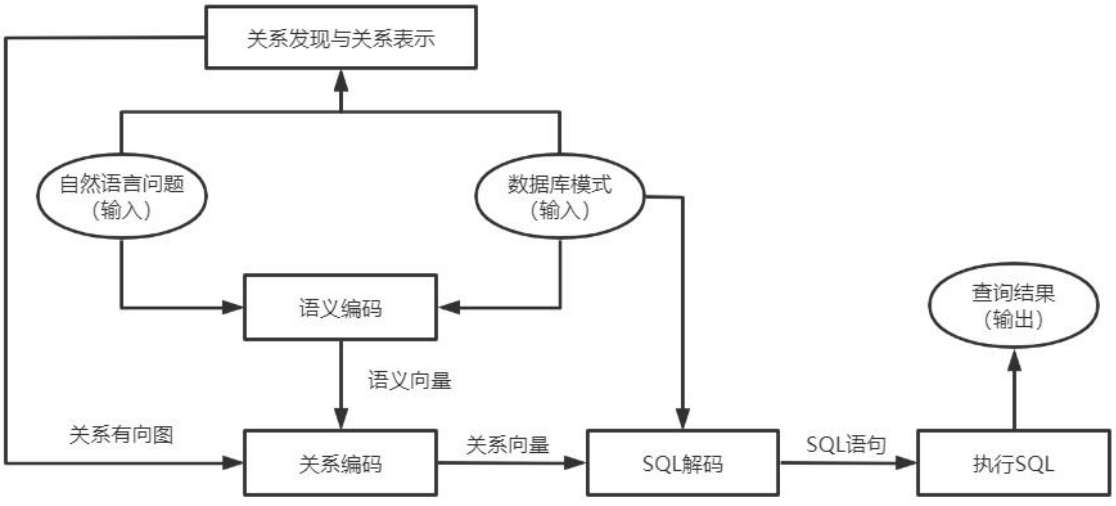

图2-1 Text-to-SQL技术框架图

2.2 BERT 预训练模型

BERT[11]是自然语言和数据库模式的语义编码过程所需用到的重要技术之一，作为一个预训练模型，由 Google 公司于 2018 年提出，当时在 11 项自然语言任务中都取得了 state-of-the-art 的结果而获得广泛的关注。BERT 基于双向 Transformer 的编码器 encoder，它的成功很大程度上归功于 Transformer 的强大。

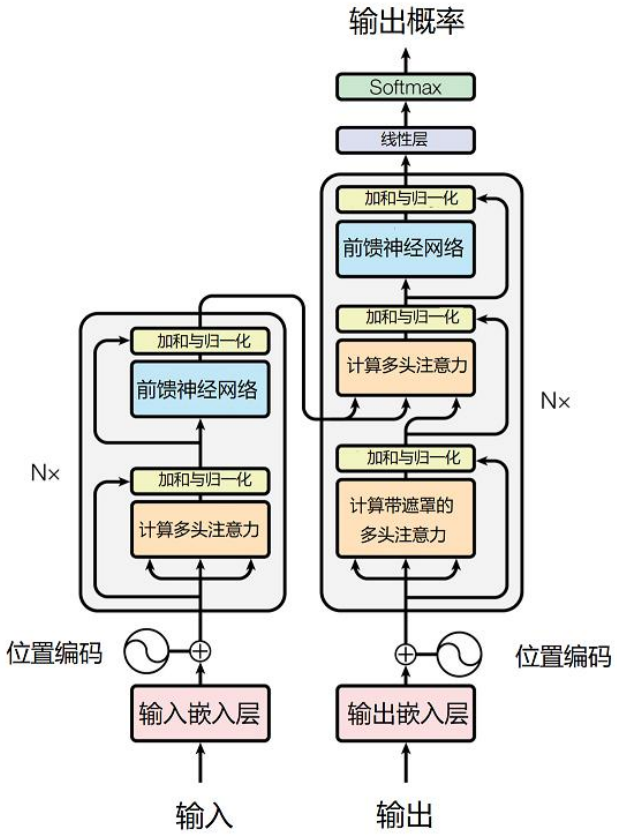

图 2-2 Transformer 架构图

Transformer[12]也是由 Google 公司提出的，图 2-2 为 Transformer 的架构图，它由基于注意力机制的多个编码器-解码器（Encoder-Decoder）组成，解码器的数量与编码器的数量相对应，每个编码器结构相同，但它们彼此独立，没有共享任何参数，由两部分组成。自然语言语句输入到编码器后，经过自注意力层，使编码器对语句中每个词进行编码时能够关注语句中的其他词，再将自注意力层计算得到的输出结果作为前馈神经网络的输入。解码器除了编码器的自注意力层和前馈神经网络之外，

还有一个编码-解码注意力层，用来关注编码后的向量。编码器和解码器核心都是多头注意力机制，使模型能捕捉到句子中不同词之间的特征关系。

图 2-3 是多头注意力结构图，公式(2-1)是注意力的通用表达式，由于这里所用的是自注意力机制，查询矩阵 Q、键矩阵 K 和值矩阵 V 关系为 Q=K=V。

$$O = \text{softmax}(QK^T)V \tag{2-1}$$

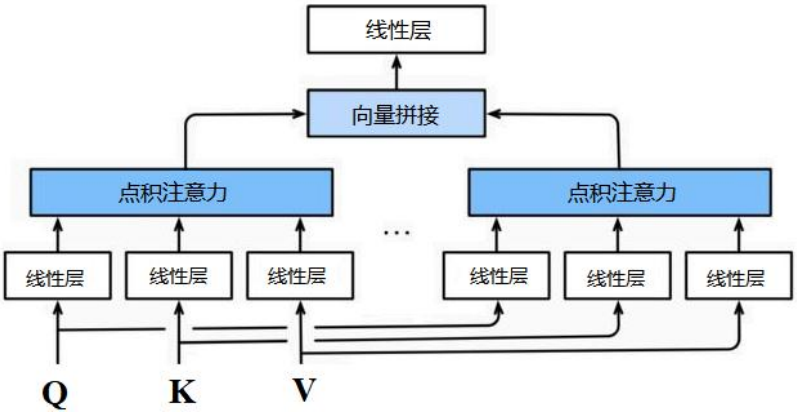

图 2-3 多头注意力结构图

多头注意力由多个独立的注意力层组成，这些层的输出共同组成最终的注意力输出。公式(2-2)、(2-3)、(2-4)是多头注意力的计算公式。其中 W 为线性层的系数矩阵，d_k 为键矩阵 K 的维度。公式中 Q 与 K 的点积除以 $\sqrt{d_k}$ 是为了让梯度更加稳定。

$$\text{Attention}(Q,K,V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{2-2}$$

$$h_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{2-3}$$

$$\text{MulHead}(Q,K,V) = \text{Concat}(h_1,...,h_h)W^O \tag{2-4}$$

在 Transformer 中，还引入了位置编码（Positional Encoding）的信息，最终的编码向量由词向量和编码向量共同组成，因此同一个词在输入语句中的不同位置上也具有不同意义。位置编码过程对每个向量进行了编号，公式(2-5)、(2-6)是位置编号的计算公式，公式(2-7)说明了最终编码向量的组成方式，其中 H 表示编码后的向量，X 表示词向量，P 表示位置向量。

$$P_{i,2j} = \sin(\frac{i}{10000^{2j/d}}) \tag{2-5}$$

$$P_{i,2j+1.} = \cos(\frac{i}{10000^{2j/d}}) \tag{2-6}$$

$$H = P + X \tag{2-7}$$

BERT 通过多层 Transformer 对输入的语句提取特征，由词编码、分片编码和位置编码来共同组成对输入语句的编码。图 2-4 为 BERT 的编码方式示意图，[CLS]和[SEP]为特殊符号，[CLS]在分类模型中用到，[SEP]代表分句符，表示其前后为两个句子。

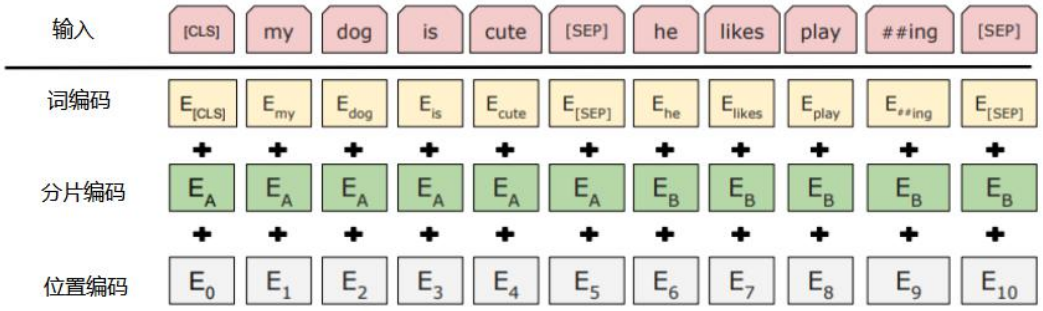

图 2-4 BERT 编码方式示意图

BERT 模型通过两个任务来进行训练：（1）遮盖语言模型（Masked Language Model，MLM）：有 15%的概率将句子中的词用[mask]遮盖起来，让模型预测这个被遮盖的词。（2）预测下个句子（Next Sentence Prediction，NSP）：将两句话输入给模型预测是否属于连接关系，这两句话从语料库中选取，有 50%的概率是相连接的两句话，以此来训练模型对两个句子之间连接关系的预测能力。

2.3 ConceptNet 知识图谱

ConceptNet[47]知识图谱是链接中文自然语言问题与英文数据库模式和常识语义推理的重要技术之一，它是一个多语言、跨领域的常识知识图谱，属于麻省理工学院（MIT）的众包项目 Open Mind Common Sense 的成果，包括 WordNet、Wiktionary 等多个常识项目中的数据，能够帮助计算机从语义的角度理解人们所用的自然语言。

ConceptNet 是一个有向图结构，其顶点为自然语言单词和短语，边为带标签的类

型和权重，权重表示该条边的可信度。图 2-5 为一个以中文“车”为顶点的语义结构图。

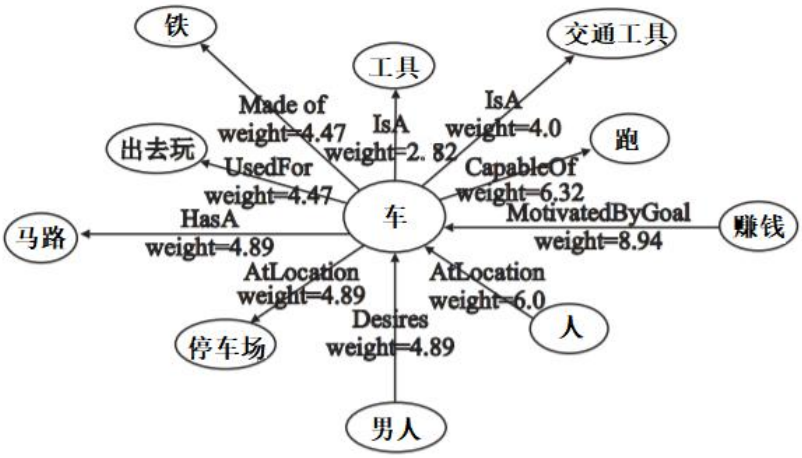

图 2-5 ConceptNet 中“车”的语义结构图

ConceptNet 包含数百种语言，可以实现跨语言的语义理解。边的关系类型有 34 种，两个顶点之间可以有多种关系，根据权重的不同赋予不同的可信度，一般选取可信度大于 1 的关系为有效关系。表 2-1 为 7 种常用的关系类型。

表 2-1 ConceptNet 中 7 种常用的关系类型

关系	意义	实例
A RelatedTo B	A 与 B 相关	学习↔博学
A IsA B	A 是 B 的子类或实例	车→交通工具
A PartOf B	A 是 B 的一部分	台湾→中国
A HasA B	A 拥有 B	鸟→羽毛
A Synonmy B	A 与 B 同义	国家↔祖国；国家↔country
A UsedFor B	A 用来 B	桥→跨海
A SimilarTo B	A 与 B 相似	猴↔猿

目前，ConceptNet 5.7 版本[6]已提供 REST 风格的 API，返回 JSON 格式数据，可

6 http://conceptnet.io/

以帮助自然语言处理中对语义的理解，在本文的中文 Text-to-SQL 任务中，ConceptNet 能帮助解决跨语言问题，也能通过常识和语义信息发现自然语言问题与数据库模式之间的链接关系。图 2-6 为返回的数据样例图。

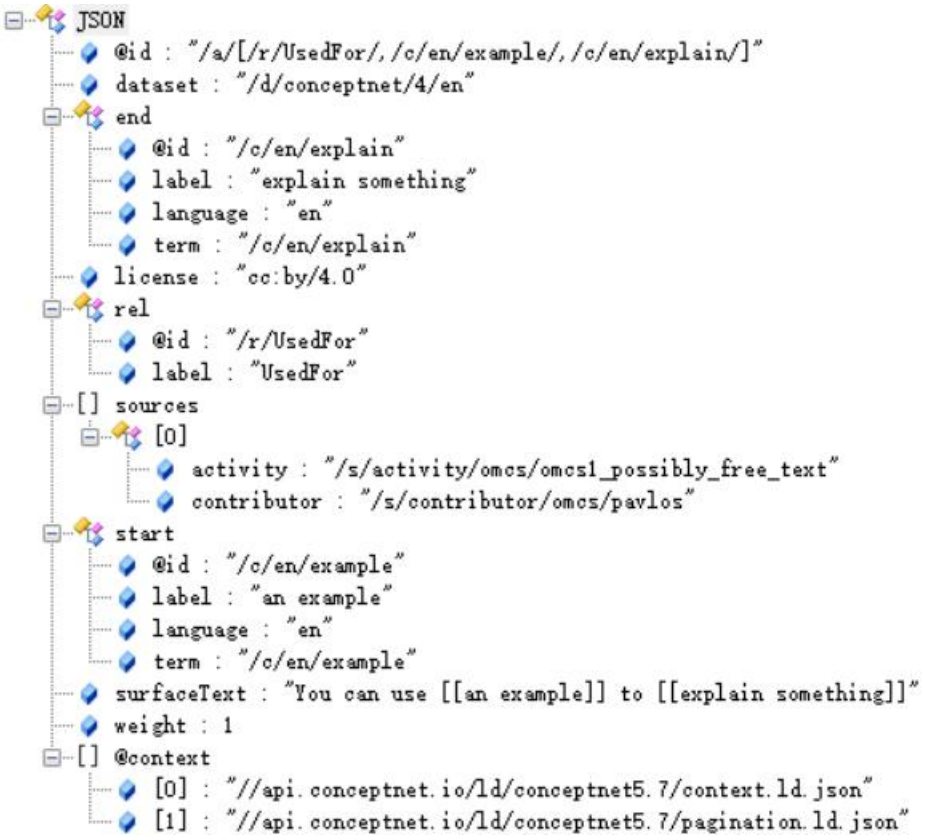

图 2-6 ConceptNet API 返回数据样例图

2.4 RAT-SQL 模型

RAT-SQL 模型[39]是基于已发现的关系有向图进行关系编码以及在解码过程中生成符合语法规则 SQL 语句的重要技术之一，是目前英文复杂 Text-to-SQL 任务上的 state-of-the-art 模型，在 Spider 数据集上达到了 65.6%准确率。

图 2-7 为 RAT-SQL 的模型结构示意图。首先，数据库模式以有向图的形式表示为G_Q。然后分别对数据列名和对应的列类型、数据表名以及组成自然语言问题的每个词进行初始编码，初始编码后的结果输入多个 Relation-Aware Transformer 层，得到最终的编码结果，如图 2-8 所示。最后将 Encoder 的编码结果输入到树形的 Decoder

中，按照 SQL 的语法规则进行解码，得到所生成的 SQL 语句。

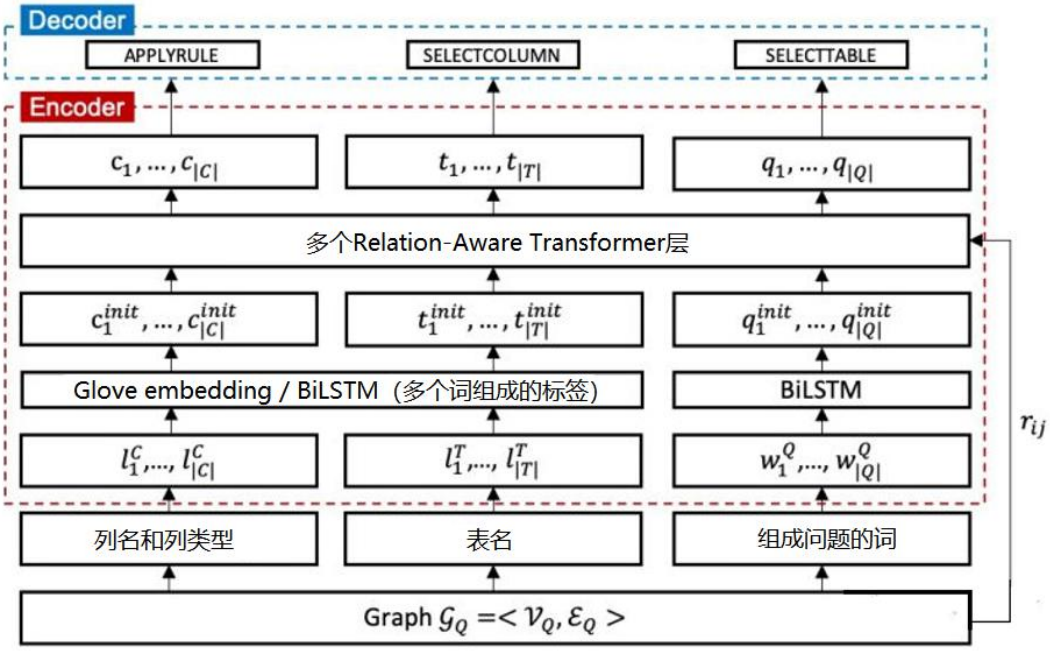

图 2-7 RAT-SQL 模型结构示意图

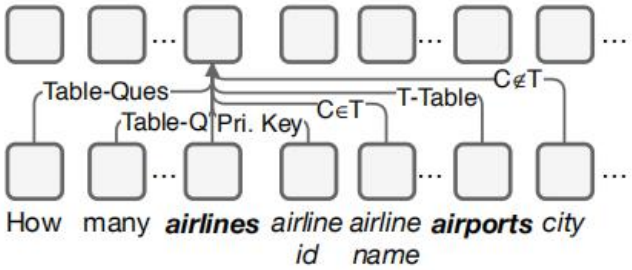

图 2-8 RAT-SQL 编码过程示意图

在 Relation-Aware Transformer 层，RAT-SQL 把显式的数据库模式关系和隐式的问题与数据库模式之间的关联关系共同进行编码，使得模型能够更好的结合数据库的结构化数据以及自然语言问题的非结构化数据，完善了模型的表达能力。

Relation-Aware Transformer 层的计算公式如下：

$$X = (c_1^{init}, \ldots, c_{|C|}^{init}, t_1^{init}, \ldots, t_{|T|}^{init}, q_1^{init}, \ldots, q_{|Q|}^{init}) \tag{2-8}$$

$$r_{ij}^K = r_{ij}^V = \text{Concat}(\rho_{ij}^{(1)}, \ldots, \rho_{ij}^{(R)}) \tag{2-9}$$

$$e_{ij}^{(h)} = \frac{x_i W_Q^{(h)} (x_j W_K^{(h)} + r_{ij}^K)^T}{\sqrt{d_z / H}} \tag{2-10}$$

$$\alpha_{ij}^{(h)} = \text{softmax}\{e_{ij}^{(h)}\} \tag{2-11}$$

$$z_i^{(h)} = \sum_{j=1}^{n} \alpha_{ij}^{(h)} (x_j W_V^{(h)} + r_{ij}^V) \tag{2-12}$$

$$z_i = (z_i^{(1)}, ..., z_i^{(H)}) \tag{2-13}$$

$$\overline{y_i} = LN(x_i + z_i) \tag{2-14}$$

$$y_i = LN(\overline{y_i} + \text{FC}(\text{ReLU}(\text{FC}(\overline{y_i})))) \tag{2-15}$$

公式(2-8)为该层的输入，由初始编码的列向量、表向量和问题向量拼接而成；公式(2-9)为偏置 r_{ij}^K 和 r_{ij}^V 的表示，由多个关系向量拼接而成，ρ_{ij}^r 表示第 r 个关系类型向量，如果它们之间没有该类型关系，则用对应维度的 0 向量表示，即没有偏置。公式(2-10)~(2-15)计算该层带偏置的自注意力，其中，FC 表示全连接层（fully-connected layer），LN 表示层归一化[48]（layer normalization）。$W_Q^{(h)}, W_K^{(h)}, W_V^{(h)} \in R^{d_x * (d_x/H)}$ 分别表示自注意力中的查询（query）、键（key）和值（value）矩阵。

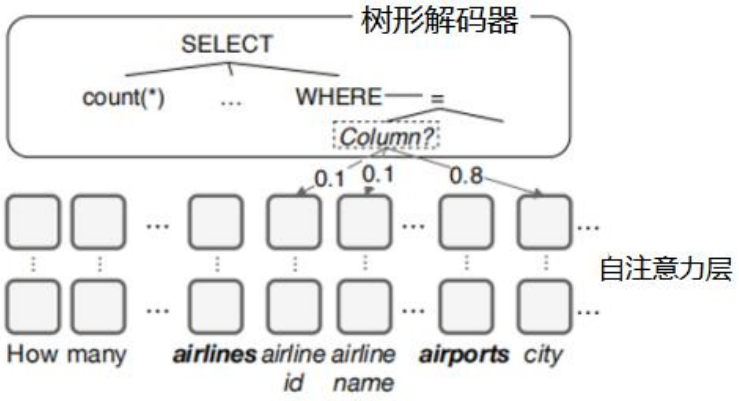

图 2-9 RAT-SQL 解码过程示意图

RAT-SQL 的解码过程使用深度优先的抽象语法树逐层进行 SQL 语句的生成，如图 2-9 所示。由于 SQL 语句有特定的语法规则，按照序列化的生成方式会出现较多的语法问题，而使用语法树进行解码能保证语法的准确性，提高整体的准确率。解码器使用 LSTM 生成一系列动作（action），包括应用语法规则生成下个 token，称作 APPLYRULE；或选择一个数据列/数据表插入当前位置，称作

SELECTCOLUMN/SELECTTABLE。

公式(2-16)为 APPLYRULE 的计算公式，其中 $g(\cdot)$ 是一个 2 层的感知器（MLP），h_t 表示 t 时刻 LSTM 的输出。公式(2-17)~(2-21)为 SELECTCOLUMN 的计算公式，使用注意力机制得到每个列的概率分布，选择概率值最大的列。SELECTTABLE 过程与 SELECTCOLUMN 同理。

$$\Pr(a_t = \text{APPLYRULE}[R] \mid a_{<t}, y) = \text{softmax}_R(g(h_t)) \tag{2-16}$$

$$\tilde{\lambda}_i = \frac{h_t W_Q^{sc} (y_i W_K^{sc})^T}{\sqrt{d_x}} \tag{2-17}$$

$$\lambda_i = \text{softmax}\{\tilde{\lambda}_i\} \tag{2-18}$$

$$\tilde{L}_{i,j}^{col} = \frac{y_i W_Q^{col} (c_j W_K^{col} + r_{ij}^K)^T}{\sqrt{d_x}} \tag{2-19}$$

$$L_{i,j}^{col} = \text{softmax}\{\tilde{L}_{i,j}^{col}\} \tag{2-20}$$

$$\Pr(a_t = \text{SELECTCOLUMN}[i] \mid a_{<t}, y) = \sum_{j=1}^{|y|} \lambda_j L_{j,i}^{col} \tag{2-21}$$

2.5 本章小结

本章主要介绍了自然语言处理和 Text-to-SQL 领域相关的理论和关键技术。说明了 Text-to-SQL 任务的总体技术框架，重点介绍了目前在自然语言处理中广泛使用且表现优秀的预训练模型 BERT，对包括 Transformer 和注意力机制在内的技术原理作说明；多语言、跨领域的常识知识图谱 ConceptNet，对常用的关系类型和 API 作举例说明；目前在英文 Text-to-SQL 任务上的 state-of-the-art 模型 RAT-SQL，对编码解码的过程和所用的计算公式作了详细说明。

本课题的工作将围绕本章所阐明 Text-to-SQL 任务中的关键理论和技术点开展。

3 中文自然语言生成复杂 SQL 的模型 RACN-SQL

本章针对中文复杂 Text-to-SQL 任务的特点，围绕第二章中阐明的理论和关键技术点，提出中文自然语言生成复杂 SQL 的模型 RACN-SQL（Relation-Aware Chinese to Structured Query Language Model）。为了使模型能够更好地捕捉自然语言问题与相关数据库模式中的语义和关系特征并依据 SQL 语法生成正确的 SQL 语句，RACN-SQL 将中文复杂 Text-to-SQL 任务分为 5 个关键步骤，分别为：（1）自然语言问题与数据库模式的语义编码；（2）构建数据库模式关系表示；（3）链接自然语言问题与数据库模式之间的关系；（4）基于上述关系进行关系编码；（5）按照 SQL 语句的语法规则生成最终的 SQL 语句。图 3-1 为 RACN-SQL 模型的语义解析流程图，关键步骤（1）~（5）已在图中标出。

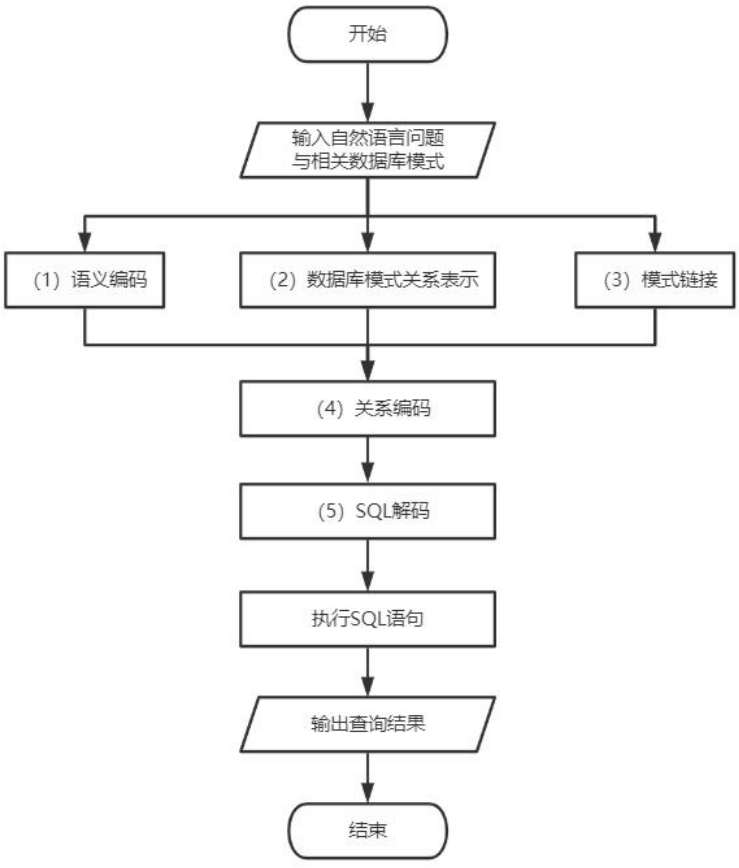

图 3-1 RACN-SQL 模型的语义解析流程图

3.1 基于 BERT 的语义编码

为了捕捉用户提出的自然语言问题与数据库模式之间的对齐关系，Text-to-SQL 任务的语义编码需要对它们同时进行编码。考虑到 BERT 预训练模型在自然语言处理任务中的优秀表现，同时提供多语言版本能够解决中文 Text-to-SQL 任务中跨语言的障碍，本文使用基于 BERT 的语义编码方案。

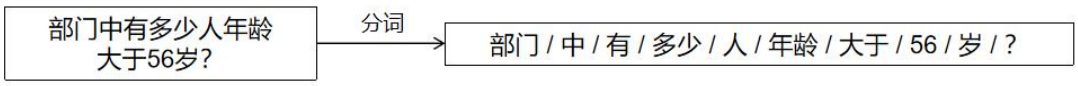

图 3-2 自然语言问题的分词过程示例

对于中文自然语言问题，首先需要进行分词操作，本课题使用目前精度较高的中文分词工具，Jieba。Jieba 分词工具能够对中文自然语言问题进行处理，返回概率最大的分词组合，但由于自然语言问题中除了汉字以外，还常包含阿拉伯数字、单位符号、标点符号等，因此对 Jieba 分词工具的输出需将上述情况分隔的子串进行组合，保持其原意不被修改。记自然语言问题为 Q，公式(3-1)表示经过分词处理后得到的子串，图 3-2 为一个自然语言问题的分词过程示例。

$$Q = (q_1, q_2, ..., q_{|Q|}) \tag{3-1}$$

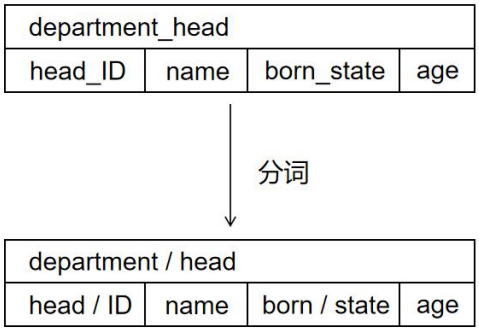

图 3-3 数据表和数据列的分词过程示例

对于数据库的列名和表名，结合工程实践的需求，是由带下划线分词符的英文组成的，因此只需根据下划线进行分词即可。公式(3-2)和(3-3)分别表示经过分词处理后得到的数据表和数据列，图 3-3 为一个数据表和数据列的分词过程示例。

$$T = (t_1, t_2, ..., t_{|T|}) \tag{3-2}$$

$$C = (c_1, c_2, ..., c_{|C|}) \tag{3-3}$$

完成分词工作后，将得到的自然语言问题、数据表以及数据列拼接起来，同时对每个数据列连接上其对应的类型。公式(3-4)为拼接得到的模型输入 X，c_{type_i} 表示第 i 个数据列的类型，$c_{type} \in \{\text{number}, \text{time}, \text{text}\}$，分别表示该列为数字类型（包括整数和浮点数）、时间日期类型、文本类型。

$$X = (q_1, q_2, ..., q_{|Q|}, t_1, t_2, ..., t_{|T|}, c_1; c_{type_1}, c_2; c_{type_2}, ..., c_{|C|}; c_{type_{|C|}}) \tag{3-4}$$

X 由中文和英文两种语言构成，因此在进行编码时，使用多语言的 BERT 预训练模型（Multilingual-Bert）。作为 BERT 模型的输入，需要再拼接上特殊符号[CLS]和[SEP]。公式(3-5)为输入到 BERT 模型中的 X_{concat} 表达式。

$$X_{concat} = [\text{CLS}]\, q_1 \ldots q_{|Q|}\, [\text{SEP}]\, t_1 \ldots t_{|T|}\, [\text{SEP}]\, c_1; c_{type_1} \ldots c_{|C|}; c_{type_{|C|}}\, [\text{SEP}] \tag{3-5}$$

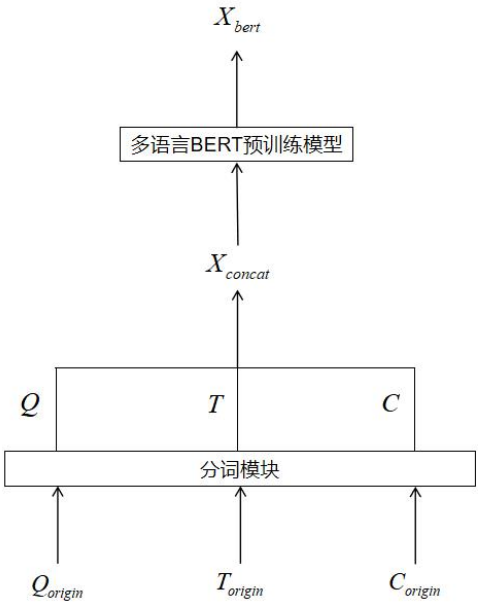

图 3-4 语义编码过程示意图

图 3-4 为语义编码过程示意图，经过分词、组合、输入多语言 BERT 预训练模型等流程，最终得到编码后的自然语言问题和数据库模式的语义向量 X_{bert}。

3.2 数据库模式关系表示

在数据库模式中存在大量已知的关系，这些关系在数据表和数据列定义时就显

式指定。对于 Text-to-SQL 任务而言，数据库模式中的关系对生成正确的 SQL 语句至关重要，决定了 SQL 语句中是否需要使用表连接、嵌套查询等操作。因此本小节的目标是将已存在的数据库模式关系表示出来。

数据库模式中的关系符合有向图的定义，以数据表和数据列作为顶点，边表示两顶点之间的关系。表 3-1 中列出了有向图中可能出现的关系类型。

通过有向图表示数据库模式中的关系之后，模型在应用注意力机制时，就能更加“关注”数据库模式中既有的关系，并且在训练过程中针对不同的关系边分配不同的注意力权重。

表 3-1 数据库模式关系有向图中的关系类型

顶点 A	顶点 B	标签	描述
列	列	同表	A 和 B 同属于一张数据表
		外键列-F	A 外键关联 B
		外键列-R	B 外键关联 A
列	表	主键-F	A 是 B 的主键
		包含-F	A 是 B 中的列，但不是 B 的主键
表	列	主键-R	B 是 A 的主键
		包含-R	B 是 A 中的列，但不是 A 的主键
表	表	外键表-F	A 中有一个列外键关联 B
		外键表-R	B 中有一个列外键关联 A
		外键表-B	A 和 B 互相外键关联

3.3 模式链接

模式链接（Schema Linking），即链接自然语言问题中的词与数据库模式中的数据表、数据列或数据值的过程。模式链接能够发现提问者需要查询的目标表和目标列并发现组成 SQL 语句的约束条件。

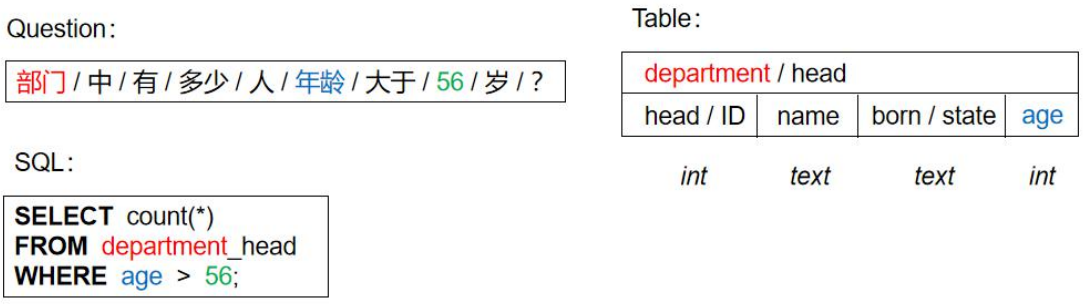

图 3-5 Text-to-SQL 任务中模式链接示例

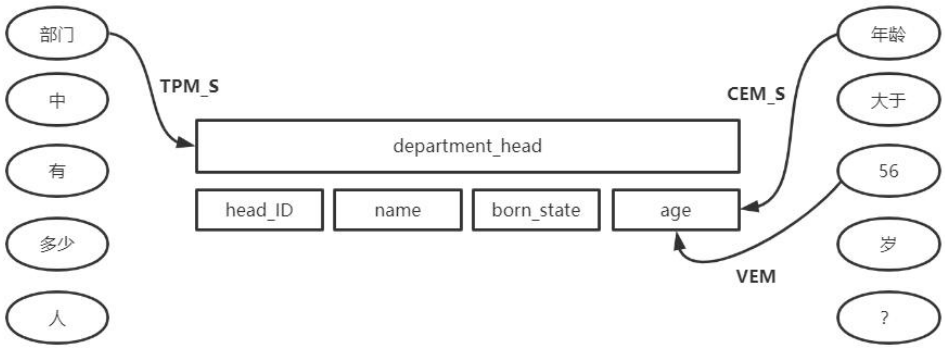

图 3-6 模式链接结果示例

图 3-5 为一个 Text-to-SQL 任务中自然语言问题、相关数据库模式、对应 SQL 语句的示例，自然语言问题和相关数据库模式都已经过 3.1 节中的分词模块处理。示例中“部门”一词与数据库模式中的数据表名 department_head 的分词后结果 department 相关联，该表出现在生成 SQL 的 FROM 子句中，说明提问者提出的问题答案需要在 department_head 这张数据表中检索；示例中“年龄”一词与数据库模式中 department_head 数据表下的 age 数据列相关联，该列出现在 WHERE 子句中作为限制条件的列；示例中“56”属于整型数值，符合 department_head 数据表中 head_ID 和 age 这两个数据列的数据类型，此外根据数据值匹配和常识网络判断，与 age 数据列相关联，因此该值作为 WHERE 子句中 age 的限制值出现。图 3-6 为该示例的模式链接结果，其中椭圆框表示自然语言问题中的词，矩形框表示数据库模式的名称。由于最终生成的 SQL 语句中包含模式链接中所发现的关联数据表、数据列以及数据值，因此模式链接过程能否准确发现自然语言中所出现的词、数据库模式之间的关联关系很大程度上影响解析准确率。

为了进行跨语言的语义和常识链接，在链接过程中，使用在第二章 2.2 节中介绍的 ConceptNet 中的关系给对应的自然语言中的词与对应的表名或列名打上关系标签。表 3-2 列出了本节模式链接中使用的 ConceptNet 关系表，第一行为同义关系，为了解决实际工程中自然语言问题为中文而数据库模式为英文的跨语言链接问题；第二~五行的关联关系目的是通过语义和常识进行链接自然语言问题中隐含的指代关系，比如中国与国家之间的 IsA 关系。在使用 ConceptNet 有向图时，同义关系只允许一条 Synonym 关系边的链接；而其他关系允许两条关系边的链接，其中一条为 Synonym 关系边，另一条为除 Synonym 之外的关系边。比如在自然语言问题中“中国”词与数据列名“country”的链接过程如图 3-7 所示，首先中文的“中国”经“Synonym”关系边与英文“China”同义链接，但“China”并不在数据库模式中，“China”继续经“IsA”关系边与英文“country”链接，“country”属于一个数据列名，完成了中文“中国”与英文“country”之间的链接。

表 3-2 模式链接中使用的 ConceptNet 关系表

关系	意义
Synonym	同义
RelatedTo	相关
IsA	子类或实例
PartOf	部分
SimilarTo	相似

图 3-7 “中国”与“country”的链接过程示例

对于表 3-2 中所列出的五种关系，同义关系的链接程度最强，而其他关系的链接程度是基于语义和常识判断，弱于同义关系。因此在对自然语言问题中词和数据库模式的关联标签中，将同义关系与其他关系分别打上不同的两类关系标签，在注意

力机制中给予不同权重，这样，模型就能给予不同链接关系以不同的关注度，提高生成 SQL 语句时选出正确表名或列名的概率。其中，“Synonym”关系在关系标签中表示为“_S”；“RelatedTo”、“IsA”、“PartOf”、“SimilarTo”关系在关系标签中表示为“_R”。

3.3.1 数据表名和数据列名的链接

对数据表名和数据列名的链接，是将自然语言问题中的每个词分别与分词后的数据表名和数据列名进行 n-gram[49]匹配。n-gram 基于统计语言模型，它通过滑动窗口的方法得到长度为 n 的片段序列，然后这些序列分别与自然语言中的词匹配。

表 3-3 数据表名和数据列名链接的关系标签类型

关系标签	描述
TEM_S	与表名同义完全匹配
TEM_R	与表名相关完全匹配
TPM_S	与表名同义部分匹配
TPM_R	与表名相关部分匹配
CEM_S	与列名同义完全匹配
CEM_R	与列名相关完全匹配
CPM_S	与列名同义部分匹配
CPM_R	与列名相关部分匹配

表 3-3 列出了 8 种数据表名和数据列名链接的关系标签类型。TEM（Table Exact Match）表示自然语言问题中的词与数据表名完全匹配；TPM（Table Partial Match）表示自然语言问题中的词与数据表名部分匹配；CEM（Column Exact Match）和 CPM（Column Partial Match）则表示列的对应关系。在进行匹配时，模型的优先级为：同义完全匹配>相关完全匹配>同义部分匹配>相关部分匹配。

图 3-8 是数据表名和数据列名的链接示例，模型将“部门”与表名“department_head”之间打上 TPM_S 标签；将“年龄”与列名“age”之间打上 CEM_S

标签。

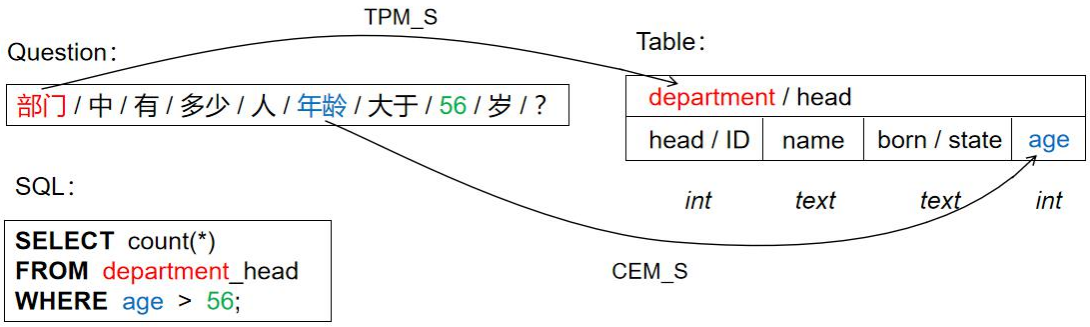

图 3-8 数据表名和数据列名的链接示例

3.3.2 数据值的链接

对数据值的链接，是将自然语言问题中的每个词与数据库中相同类型的列中所存储的数据值进行链接，或由 ConceptNet 中的常识关系边进行链接。表 3-4 列出了数据值链接的关系标签类型。对于图 3-6 的示例，数据列“country”存在“中国”数据值时，则模型将“中国”与列名“country”之间打上 VEM（Value Exact Match）标签；数据列“country”不存在“中国”数据值时，由 ConceptNet 的 Synonmy 和 IsA 关系边完成链接，则模型将“中国”与列名“country”之间打上 VRM（Value Related Match）标签。

表 3-4 数据值链接的关系标签类型

关系标签	描述
VEM	值出现在数据列中
VRM	值与数据列名相关

在进行关系标签为 VEM 的链接时，首先根据自然语言问题中词的类型选择出与之相同的数据列，类型包括数字型 number、日期型 date、文本型 text，然后检查该词是否作为数据值出现在数据列中，如果出现则将自然语言中对应的词与出现的数据列之间用 VEM 关系标签链接。

在进行关系标签为 VRM 的链接时，由于链接的对象分别是中文的词与英文的数

据列名，因此在 ConceptNet 中允许两条关系边的链接，其中一条为 Synonym 关系边，另一条为除 Synonym 之外的关系边。根据关系边得到相关联的词，与数据列进行链接，VRM 链接要求关联词与数据列完全匹配。

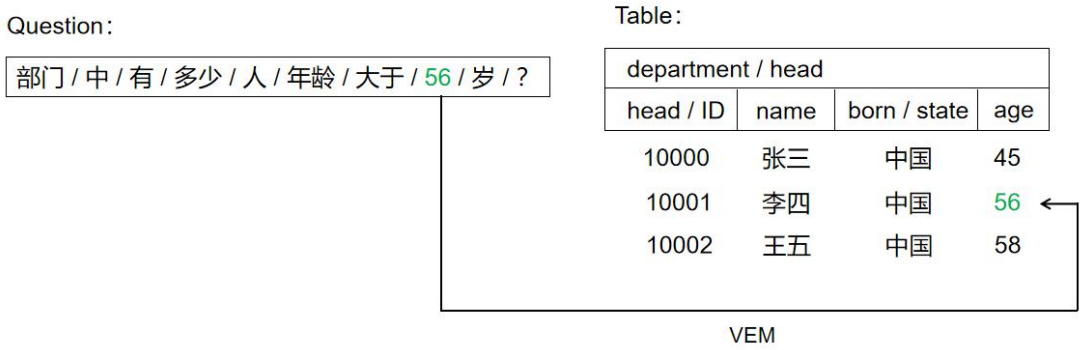

图 3-9 数据值链接 VEM 关系示例

图 3-9 为数据值链接中关系标签为 VEM 的示例，自然语言问题中“56”属于数字（整型），与数据表 department_head 中数据列 age 的类型一致，且在该数据列的所有数据值中存在“56”，因此自然语言问题中的“56”与数据列 age 链接 VEM 关系。

图 3-10 为数据值链接中关系标签为 VRM 的示例，自然语言问题中“中国”在数据库模式中不存在这个数据值。由 ConceptNet 中的两条关系边得到关联词 country，与数据表 goods 中数据列 country 完全匹配，因此自然语言问题中的“中国”与数据列 country 链接 VRM 关系。

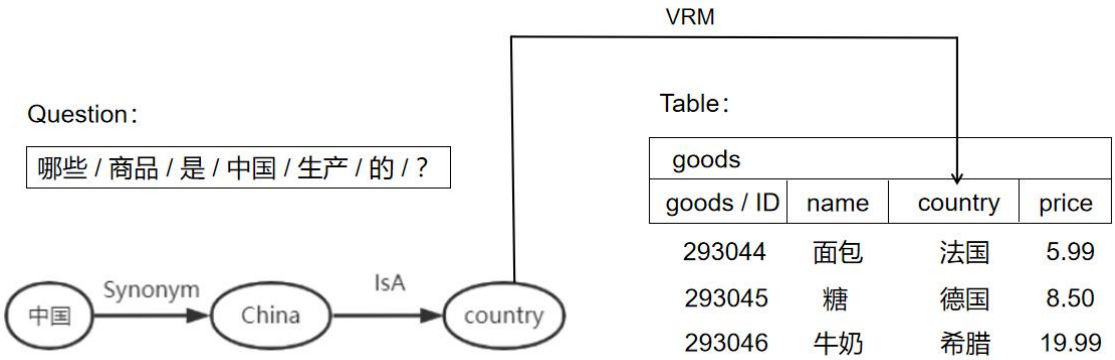

图 3-10 数据值链接 VRM 关系示例

3.4 基于 Transformer 的关系编码

在前面的小节中，已经将自然语言问题和数据库模式进行了语义编码，也将数据库中的模式关系、与自然语言中词的链接关系用有向图表示出来。本节将基于 Transformer 对这些关系进行关系编码，得到自然语言问题和数据库模式的关系向量表示。

图 3-11 为基于 Transformer 的关系编码过程示意图，X_{bert} 为 3.1 节语义编码的输出，Transformer 首先进行位置编码，然后计算多头注意力，进行加和与层归一化后输入到一个前馈神经网络中，最后再经加和与层归一化得到关系编码的输出 X_{encode}。

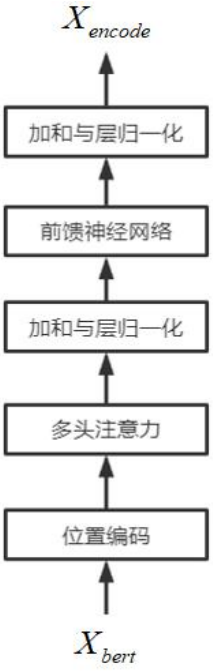

图 3-11 基于 Transformer 的关系编码过程示意图

公式(3-6)和(3-7)分别是偶数位置和奇数位置的位置编码向量计算方式，公式(3-8)中 X_{pos} 为位置编码的输出，为原向量与位置向量的加和。

$$P_{i,2j} = \sin(\frac{i}{10000^{2j/d}}) \tag{3-6}$$

$$P_{i,2j+1.} = \cos(\frac{i}{10000^{2j/d}}) \tag{3-7}$$

$$X_{pos} = P + X_{bert} \tag{3-8}$$

Transformer 的核心是多头注意力机制，使计算机能够对输入中的关键信息给予更多“关注”，图 3-12 是注意力计算过程示意图，注意力的核心思想是将一个查询（query）映射到一组键值对（key-value）上。计算方法是：（1）计算查询与每个键的相关性得到对值的权重；（2）将权重进行 softmax 归一化处理得到每个值的权重系数；（3）将值与对应权重系数相乘累加后得到注意力计算结果。

在 Text-to-SQL 任务的注意力机制中，需要模型能够捕捉数据库模式关系和模式链接关系，即 3.2 和 3.3 小节中所有可能出现的关系标签。因此定义r_{ij}，表示两

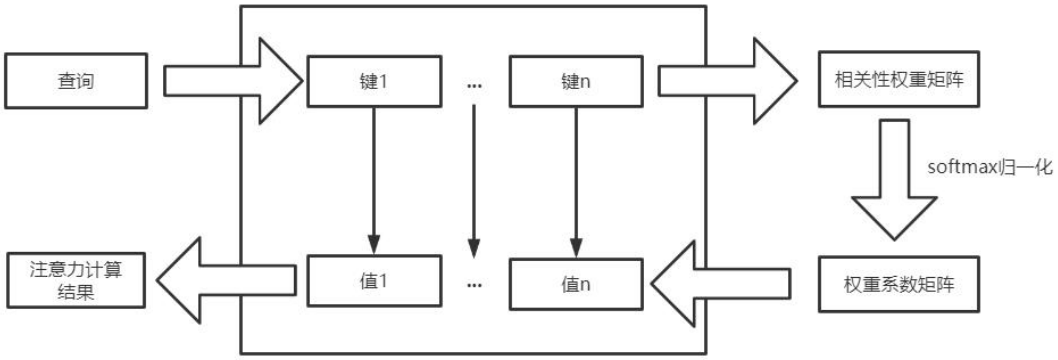

图 3-12 注意力计算过程示意图

个输入i, j的关系向量，如公式(3-9)所示，该向量由多个关系α_{ij}拼接而成，R 由所有关系标签类型的数量决定，对于i, j之间不存在的关系，α_{ij}用相应维度的 0 向量代替。

$$r_{ij} = \text{Concat}(\alpha_{ij}^{(1)},...,\alpha_{ij}^{(R)}) \tag{3-9}$$

$$Q_i^{(h)} = x_i W_Q^{(h)} \tag{3-10}$$

$$K_{ij}^{(h)} = x_j W_K^{(h)} + r_{ij}^K \tag{3-11}$$

$$V_{ij}^{(h)} = x_j W_V^{(h)} + r_{ij}^V \tag{3-12}$$

$$A_{ij}^{(h)} = \text{softmax}(\frac{Q_i^{(h)}(K_{ij}^{(h)})^T}{\sqrt{d_{\text{model}}/H}})V_{ij}^{(h)} \tag{3-13}$$

$$\text{head}_i^{(h)} = \sum_{j=1}^{n} A_{ij}^{(h)} \tag{3-14}$$

$$O_i = \text{Concat}(\text{head}_i^{(1)}, \text{head}_i^{(2)}, ..., \text{head}_i^{(H)}) W^O \tag{3-15}$$

公式(3-10)~(3-12)分别为注意力机制中 Q、K、V 的计算公式，通过增加关系向量的r_{ij}，模型对数据库模式存在的关系和模式链接中发现的关系给予更多权重，在训练过程中，这些权重值由反向传播过程进行改变，最终每个关系便签会被给予不同的权重值，反映了不同关系标签的重要程度。

公式(3-13)~(3-14)是一个独立的注意力计算公式，RACN-SQL 模型中使用点积注意力。公式(3-15)是多头注意力O_i的计算公式，该值由多个独立的注意力值拼接并乘以系数矩阵而得到。

如公式(3-16)所示，在完成多头注意力的计算后，将多头注意力结果与输入加和并进行层归一化。公式(3-17)是前馈神经网络的计算公式，输入y_i经过线性层后经 ReLU 激活函数处理，再经过一线性层得到输出z_i。公式(3-18)和(3-19)中，将前馈神经网络的输出进行加和与归一化后得到关系编码的输出X_{encode}。

$$y_i = \text{LayerNorm}(x_i + O_i) \tag{3-16}$$

$$z_i = \text{Linear}(\text{ReLU}(\text{Linear}(y_i))) \tag{3-17}$$

$$X_i = \text{LayerNorm}(y_i + z_i) \tag{3-18}$$

$$X_{encode} = \{X_i\}_{i=1}^{n} \tag{3-19}$$

3.5 基于语法树的 SQL 语句解码

SQL 语句解码是由关系编码后的向量X_{encode}生成 SQL 语句的过程。由于 SQL 的约束语法本质上是树形结构，使用基于语法树的解码方式能够生成具有复杂结构的 SQL 语句，同时保证生成的 SQL 语句语法均符合规范。

基于语法树的 SQL 语句解码示例如图 3-13 所示，语法树的根节点为 root，按照深度优先的顺序生成结点，生成的结点可能有两大类：（1）生成 SQL 语法关键字或结束符，称为 GenGrammar；（2）生成数据表名、数据列名，分别称为 GenTable

和 GenCol。每个结点的后继结点可能的类型由 SQL 语法决定，在解码时，通过 LSTM 来生成解码序列。

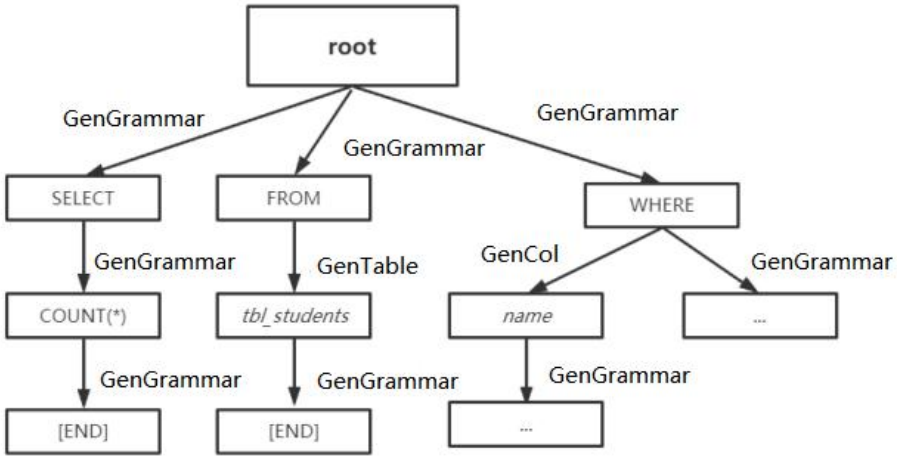

图 3-13 基于语法树的 SQL 语句解码示例

在 LSTM 的 t 时刻，模型对细胞状态 s_t 和输出 o_t 的更新如公式(3-20)所示，其中 action 表示 GenGrammar、GenTable 或 GenCol； action_{pt} 表示当前结点的父结点的行为； z_t 是上下文向量，由上一时刻输出 o_{t-1} 经过多头注意力计算而得； o_{pt} 表示当前结点的父结点时刻的输出， type_t 表示当前结点的类型向量。

$$s_t, o_t = \text{LSTM}([\text{action}_{t-1}, \text{action}_{pt}, z_t, o_{pt}, \text{type}_t], s_{t-1}, o_{t-1}) \tag{3-20}$$

模型的目标是最大化正确 SQL 语句的概率值，如公式(3-21)所示，其中 action_{pre} 表示所有之前的行为序列。

$$P_{decode} = \prod_t \text{Prob}(\text{action}_t \mid \text{action}_{pre}, X_{encode}) \tag{3-21}$$

语法树的每个分支都最终以[END]结束，当所有分支都结束时，按照深度优先的顺序遍历语法树，即可生成符合语法规则的 SQL 语句，将 SQL 语句交由数据库执行引擎即可返回 SQL 查询结果。

3.6 本章小结

本章提出了中文自然语言生成复杂 SQL 的模型 RACN-SQL。针对中文复杂

Text-to-SQL 任务的特点，结合语义编码、关系编码、数据库模式关系和自然语言问题与数据库模式链接关系等，模型将整个任务分为五个关键步骤，从设计流程、技术要点、所用公式等方面进行详细介绍，并给出了相关示例。

语义编码阶段，将分词后的自然语言问题和数据库模式组合，加入数据列的类型，组合后的结果作为多语言 BERT 预训练模型的输入，得到语义编码的向量输出；数据库模式关系表示阶段，构建有向图，顶点是数据表或数据列，边表示了顶点之间的关系；模式链接阶段，分为数据表和数据列的链接、数据值的链接两个部分，前者利用 n-gram 统计语言模型对分词后的数据表名和数据列名进行关联，后者利用不同数据列的字符和类型特征、数据列中的数据值、语义和常识特征来进行关联；关系编码阶段，基于 Transformer 对语义向量进一步训练；SQL 语句解码阶段，基于语法树，使用 LSTM 来生成解码行为序列，按照语法树深度优先顺序来生成每个结点。

4 实验与分析

本章将通过实验验证本文所提出的中文自然语言生成复杂 SQL 的模型 RACN-SQL 在中文 Text-to-SQL 任务上的有效性，分析模型的优势与不足。在对实验软硬件环境、使用的数据集、对比模型、各模型参数设置、评价指标等基本信息进行介绍后，对实验结果数据进行分析和评估。

4.1 实验环境

表 4-1 所示为本文实验运行的硬件配置环境，表 4-2 所示为本文实验所用软件版本环境。

表 4-1 实验硬件配置环境

名称	配置
CPU	Intel Core I9-10920X @ 3.50GHz 12 核
GPU	NVIDIA GeForce RTX 3090 24G * 2
内存	128 GB
操作系统	Ubuntu 18.04 64 位

表 4-2 实验软件版本环境

名称	版本号
Python	3.8.5
Pytorch	1.7.1+cu110
CoreNLP	3.9.2

实验中对 RACN-SQL 模型以及其他对比模型的训练与预测过程都在 Ubuntu 操作系统的服务器上完成，实现语言为 Python，使用 Pytorch 作为深度学习框架，训练和测试都在 GPU 上进行。

4.2 实验数据集

实验使用的数据集为2019年发布的中文复杂SQL数据集CSpider[45]，该数据集是由目前英文复杂Text-to-SQL领域广泛使用的Spider数据集经过人工翻译和校对得到的。作者为了与工程实践场景中的标准保持一致，在对自然语言问题翻译的同时，保留了英文的数据库模式命名，增加了跨语言难度，但也使得在CSpider上表现优秀的模型更能体现实用性，实验结果更具说服力。

表4-3为CSpider中数据量统计表，表4-4为SQL语句的关键字统计表。由于CSpider数据集未公开测试集，因此表中只统计了训练集和验证集的数据。CSpider数据集中共有166个数据库，涉及138个细分领域，属于通用数据集。从SQL语句关键字统计表可知，CSpider涉及了所有SQL语句中的关键字和复杂结构，并且复杂结构占总数据量比例较高，符合实际中用户的需求。

CSpider还根据SQL语句中出现关键字的多少以及结构的复杂度为依据将每个问题分为了简单（easy）、中等（medium）、困难（hard）、极难（extra hard），方便对实验结果分析，每个难度的问题示例见1.3节1.3.4小节的介绍。

表4-3 CSpider数据集数据量统计表

	问题数量	数据库数量	SQL语句数量
训练集	8659	146	4720
验证集	1034	20	547
全部	9693	166	5267

表4-4 CSpider数据集SQL语句关键字统计表

	JOIN	GROUP BY	ORDER BY	INTERSECT	UNION	EXCEPT	LIMIT	嵌套
训练集	6142	1968	1806	250	67	209	1168	4251
验证集	522	279	241	40	13	31	193	576
全部	6664	2247	2047	290	80	240	1361	4827

4.3 对比模型

本文选取了 Text-to-SQL 领域中的经典模型与目前表现优秀的模型作为对比模型。

（1）SQLNet 模型

SQLNet 模型[27]作为 Text-to-SQL 领域的经典模型，由 Xu 等人提出，对 Seq2SQL 进行了改进。为了解决 WHERE 子句中多个约束条件的顺序性问题，使用序列到集合（Seq2Set）结构和列注意力机制取代了 Seq2SQL 中使用强化学习预测 WHERE 子句的约束条件。

（2）SyntaxSQLNet 模型

SyntaxSQLNet 模型[46]是 CSpider 数据集的 baseline 模型，由 Yu 等人提出，针对复杂 Text-to-SQL 任务。由于 SQL 结构复杂，模型使用动态的 Schema 解码方案来管理 SQL 各部分之间存在的逻辑关系，在预测 Token 时，根据不同的位置，Token 的选择范围和上下文的依赖条件也不同。通过动态管理，模型能够准确地解码复杂结构的 SQL 语句。

（3）EditSQL 模型

EditSQL 模型[37]由 Zhang 等人提出的利用人机交互过程不断对所生成 SQL 语句进行修改的方法来提高所生成 SQL 语句的准确率。作者观察到相似的自然语言问题在语言上有依赖关系，并且它们所生成的 SQL 语句往往有重叠，因此在 EditSQL 模型中，将 SQL 视为序列，并使用基于 BERT 的问题-表编码器和表感知解码器，在 Token 级别生成关键字。

（4）RYANSQL 模型

RYANSQL 模型[38]由 Choi 等人于 2020 年提出，是在复杂 Text-to-SQL 上表现优秀的基于草图的槽填充模型。该模型中，作者提出了对复杂 SELECT 语句的草图以及一个填充的网络结构，并使用位置码来递归地使用槽填充方法预测嵌套查询，另外，使用两个针对模型输入的启发式方法来进一步提高模型的效果。

（5）RAT-SQL 模型

RAT-SQL 模型[39]作为目前英文复杂 Text-to-SQL 任务上的 state-of-the-art 模型，

是本文重点进行对比的模型，其核心是通过关系感知的注意力机制将自然语言问题和数据库中的关系用向量表示出来。

4.4 评价指标

对于 Text-to-SQL 任务来说，目前主流的研究工作以及官方榜单上都以逻辑准确率来评估模型的有效性。逻辑准确率是指通过模型所生成的 SQL 语句与正确的 SQL 语句在结构、关键字、数据值上完全一致。公式(4-1)为逻辑准确率的计算公式，其中 N_{lf} 是逻辑上预测正确的问题个数， N 是全部问题的个数。

$$Acc_{lf} = \frac{N_{lf}}{N} \tag{4-1}$$

另外，还有一种评价指标如公式(4-2)所示，称为执行准确率，其中 N_{ex} 是执行结果预测正确的问题个数， N 是全部问题的个数。执行准确率只要求最终由数据库执行引擎返回的结果与正确 SQL 语句所执行的结果一致，常导致最终准确率偏高。

$$Acc_{ex} = \frac{N_{ex}}{N} \tag{4-2}$$

在对最终结果的评估上，本文选取主流的逻辑准确率作为准确率评价指标，结合 CSpider 数据集中将数据分为了 4 个难度等级：简单、中等、困难、极难，因此在实验中分别根据这 4 个难度的逻辑准确率，来评估模型对不同难度问题的处理能力。

为了评估模型对不同 SQL 组成成分的预测能力，实验中，还对组成复杂 SQL 语句进行了拆分，包括：SELECT 子句、WHERE 子句、GROUP BY 子句（包括 HAVING）、ORDER BY 子句、AND/OR、IUEN（INTERSECT、UNION、EXCEPT 和嵌套）、SQL 关键字集合。

对于不同组成成分的预测，评价指标为准确率、精准率、召回率和 F1 值。公式(4-3)、(4-4)、(4-5)、(4-6)是准确率、精准率、召回率和 F1 值的计算公式，其中 TP 和 TN 分别表示标签为正类和负类且预测结果正确的数量；FP 和 FN 分别表示标签为正类和负类且预测结果错误的数量。

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{4-3}$$

$$Prec = \frac{TP}{TP + FP} \tag{4-4}$$

$$Rec = \frac{TP}{TP + FN} \tag{4-5}$$

$$F_1 = \frac{2TP}{2TP + FP + FN} \tag{4-6}$$

4.5 实验结果与分析

实验中，本文所提出的 RACN-SQL 模型的参数设置如下：

（1）batch_size：12

（2）BERT 学习率：10^{-5}

（3）多头注意力数量：8

（4）Transformer 层数：8

（5）Transformer 学习率：10^{-4}

（6）解码器 LSTM 隐藏层特征数：512

表 4-5　不同模型在 CSpider 数据集上准确率

	Easy	Medium	Hard	Extra Hard	全部
SQLNet	0.068	0.000	0.000	0.000	0.016
SyntaxSQLNet	0.336	0.132	0.190	0.006	0.170
EditSQL	0.268	0.091	0.086	0.029	0.123
RYANSQL	0.597	0.415	0.324	0.227	0.413
RAT-SQL	0.572	0.470	0.414	0.247	0.449
RAT-SQL(Aug)	0.672	0.534	0.460	0.271	0.512
RACN-SQL	**0.708**	**0.573**	**0.494**	**0.288**	**0.545**

其他对比模型的参数按照原论文中所给出的默认值进行设置。为了使模型能够解决 CSpider 数据集中跨语言的问题，在 RYANSQL 和 RAT-SQL 中，将原版模型使用的 BERT 预训练模型修改为多语言版本进行实验；由于 RAT-SQL 模型的模式链接使

用字符串匹配方案，因跨语言的问题失效，本文利用 ConceptNet 中的同义关系边完善了模式链接过程，对模型在中文数据集 CSpider 上的表现进行改进，改进后的 RAT-SQL 模型用 RAT-SQL(Aug)表示。

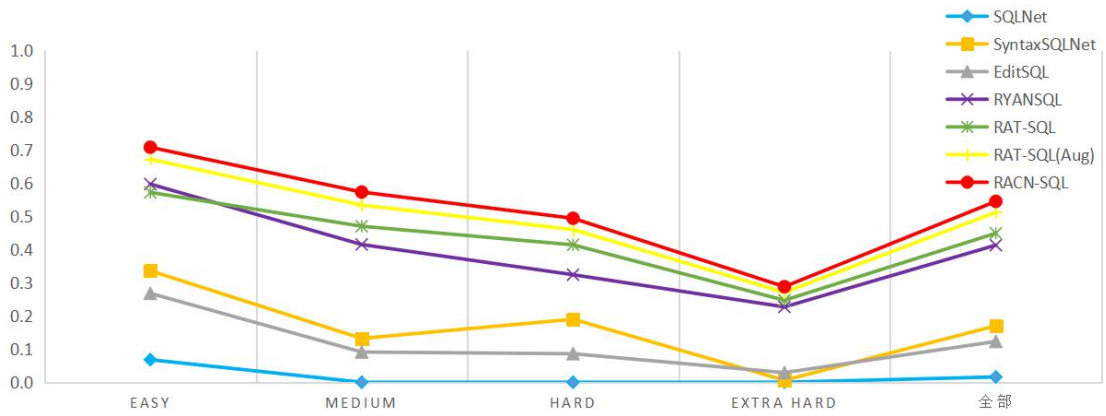

图 4-1 不同模型在 CSpider 数据集上的准确率

表 4-5 所示为不同模型在 CSpider 数据集上准确率，图 4-1 为对应折线图，按照 CSpider 数据集的难度划分，分别统计了简单（Easy）、中等（Medium）、困难（Hard）、极难（Extra Hard）以及全部的最终生成 SQL 语句的逻辑准确率，与目前 Text-to-SQL 任务上的经典模型和表现优秀的模型相比，本文所提出的 RACN-SQL 模型在各个难度的问题上准确率都有所提高，在低难度的问题上，准确率提高的幅度更大。

RYANSQL 作为一个基于草图的槽填充模型，由于低难度的问题 SQL 语句结构较简单，需要填充的槽较少，在低难度的问题上表现更加优秀。作为英文数据集 Spider 上的 state-of-the-art 模型，RAT-SQL 的解码过程基于语法树，对更为复杂的问题处理能力更强。另外，对 RAT-SQL 模型源码分析发现，RAT-SQL 中对自然语言与数据库模式的链接过程是基于字符串匹配来完成的，不能处理 CSpider 中的跨语言的特性，导致编码过程中发现的关系偏少，造成了一定准确率的降低，因此本文对原版模型适当改进，使作者在原论文中的处理方案能够跨语言应用于 CSpider 数据集，改进后的模型相比于原版模型准确率有显著提高。

SQLNet、SyntaxSQLNet 以及在英文 Text-to-SQL 任务上表现较好的 EditSQL 模型，在中文 Text-to-SQL 任务中表现并不理想，原因在于这些模型的架构相对简单，但 CSpider 数据集中存在大量复杂的结构和跨语言特性，经典的方法难以捕捉这些特

征并生成具有复杂结构的 SQL 语句。

RACN-SQL 模型在中文复杂 Text-to-SQL 任务上取得了较好的成绩，但还是可以发现对于极难的问题，RACN-SQL 模型的提升比较有限，准确率较低。纵观目前的主流模型，对高难度问题的处理效果都不佳。这种高难度的问题中存在大量复杂的关系和结构，同时难以直接从自然语言问题的表达以及数据库模式推断出来，因此对模型的语义理解、关系发现、减弱噪声、准确解码的能力都提出更高的要求，这也是目前 Text-to-SQL 领域的一个难点。

表 4-6　RACN-SQL 模型对 SQL 各成分的预测数据

	准确率	精准率	召回率	F1
SELECT	0.807	0.807	0.807	0.807
WHERE	0.609	0.609	0.609	0.609
GROUP BY	0.769	0.769	0.743	0.756
ORDER BY	0.732	0.732	0.726	0.729
AND/OR	0.969	0.968	0.988	0.978
IUEN	0.321	0.321	0.321	0.321
KEYWORDS	0.855	0.855	0.851	0.853

表 4-7　RAT-SQL(Aug)模型对 SQL 各成分的预测数据

	准确率	精准率	召回率	F1
SELECT	0.782	0.782	0.782	0.782
WHERE	0.608	0.610	0.604	0.607
GROUP BY	0.730	0.730	0.695	0.712
ORDER BY	0.690	0.689	0.705	0.697
AND/OR	0.965	0.965	0.987	0.976
IUEN	0.400	0.400	0.436	0.417
KEYWORDS	0.845	0.845	0.847	0.846

表 4-6 是 RACN-SQL 模型对 SQL 各成分的预测数据，表 4-7 是 RAT-SQL(Aug)模型对 SQL 各成分的预测数据，其中 IUEN 代表 INTERSECT、UNION、EXCEPT 和嵌套查询 4 种操作，KEYWORDS 表示 SQL 关键字的集合。图 4-2 是 RACN-SQL 与 RAT-SQL(Aug)对 SQL 各成分预测结果的准确率和 F1 值折线图。

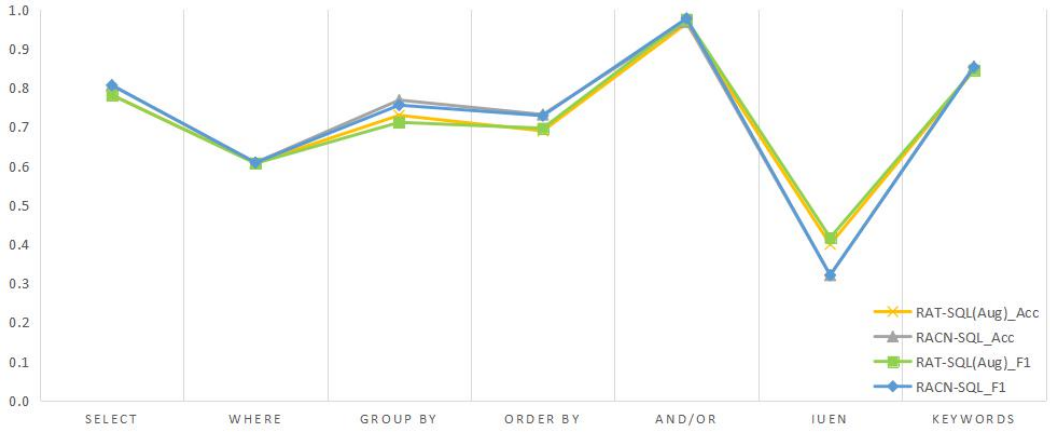

图 4-2　RACN-SQL 与 RAT-SQL(Aug)对 SQL 各成分预测结果的准确率和 F1 值

分析统计结果可知，RACN-SQL 模型对于 SELECT、GROUP BY、ORDER BY、AND/OR、KEYWORDS 部分的预测效果较好，而对 WHERE 和 IUEN 的预测效果较差，说明模型通过训练对自然语言问题中表达的检索对象、分组操作、排序操作以及各 SQL 语句成分的编解码能力好于对条件约束、交并差集以及嵌套查询，原因可能是关系发现环节中由于遗漏少部分关系或产生部分噪声，干扰了关系编码环节的编码效果，另外 IUEN 的数据库操作在原自然语言问题中缺少明确的信号，导致模型很难直接判断出是否需要进行 IUEN 操作，造成准确率低。RACN-SQL 模型对 SQL 语句成分中 SELECT、GROUP BY、ORDER BY 的预测效果好于 RAT-SQL(Aug)，IUEN 的预测效果差于 RAT-SQL(Aug)，WHERE 和 KEYWORDS 的预测效果相差不大。可见，RACN-SQL 模型对于 IUEN 这类复杂操作的编解码能力还有所欠缺，也造成了高难度问题的预测准确率低。

为了进一步探究 RACN-SQL 中不同模块对最终生成 SQL 语句准确率的影响，对去除 RACN-SQL 模型不同模块后的效果进行了消融实验，表 4-8 为去除 RACN-SQL 模型的不同模块后的准确率，图 4-3 为对应柱形图。

表 4-8 去除 RACN-SQL 模型不同模块后的准确率

	Easy	Medium	Hard	Extra Hard	全部
RACN-SQL	0.708	0.573	0.494	0.288	0.545
- 数据库模式关系	0.636	0.527	0.425	0.218	0.485
- 不同关系标签	0.716	0.545	0.454	0.218	0.517
- 语义和常识关系	0.732	0.555	0.391	0.265	0.522
- 模式链接	0.692	0.466	0.402	0.235	0.472

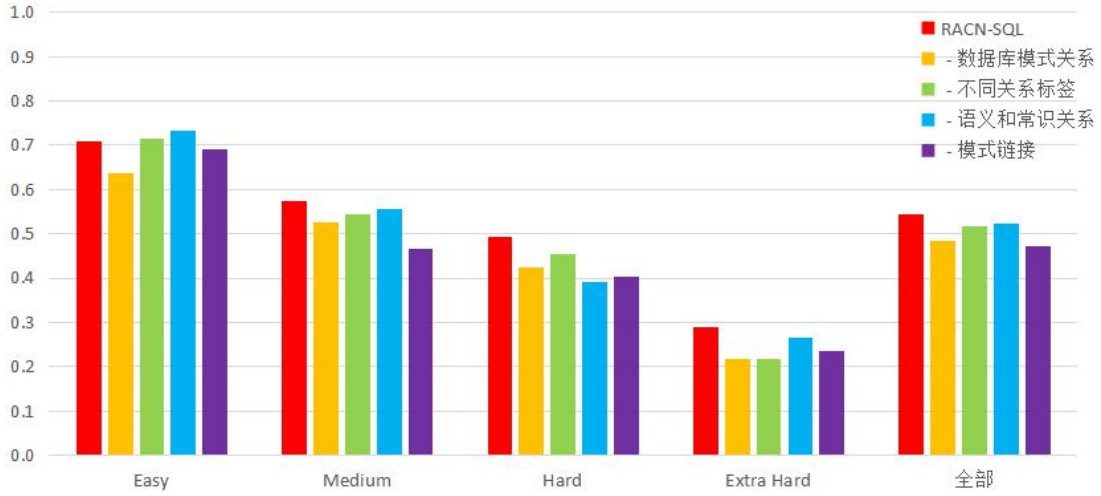

图 4-3 去除 RACN-SQL 模型不同模块后的准确率

在不考虑数据库模式关系和去除模式链接后，在各难度问题上，模型准确率均有较大幅度的降低，由此可见，Text-to-SQL 任务不同于序列到序列的简单问答任务，数据库中已有的模式关系以及模式链接中通过字符特征、语义特征和常识特征发现的关系等结构化数据的特征对于最终解析结果有很大影响，精确表示出这些结构化数据特征并合理过滤其中的噪声干扰是提高 Text-to-SQL 任务准确率的重点之一。在去除不同关系标签划分（即所有模式链接中的关系都使用同一标签训练）以及不考虑语义和常识关系后，整体的准确率略有降低，但发现对于简单（Easy）难度的问题，准确率反而有所提高，由此可见这些不同关系标签所代表的不同注意力权重以及语义和常识信息能够一定程度提高复杂 Text-to-SQL 任务解析的准确率，这是由于高难度的问题中，往往隐含有需要进行语义和常识推理的关系，这些关系特征难以

从字面上推断出来，但对于简单的问题，自然语言问题中常精确指出了所需查询的数据表名、数据列名以及数据值，通过简单的字符串匹配就能够推理出正确的 SQL 语句，这时，通过语义和常识得到的关系特征反而成为了噪声，对解析准确率会造成一定影响。

4.6 本章小结

本章通过对比实验、SQL 语句各成分的预测实验和消融实验综合验证本文所提出的 RACN-SQL 模型的有效性，并分析模型的优势与不足，指出中文复杂 Text-to-SQL 模型设计的难点和未来研究需要关注的方向。

本章首先对实验的硬件环境及软件环境进行简单介绍；接着介绍了实验所用的数据集 CSpider，说明该数据集的特点及难点；然后对实验中选取的对比模型进行介绍，选取了 Text-to-SQL 任务上的经典模型和目前表现优秀的代表性模型，说明模型的提出者和核心的设计思想及原理；同时，说明了实验中所用的评价指标，对比实验在不同难度的问题上比较模型的准确率，给出了逻辑准确率和执行准确率的计算公式，选取目前主流且更为严谨的逻辑准确率作为对比实验的评价指标；在对 SQL 语句不同组成成分的实验中，分别选取准确率、精确率、召回率和 F1 值作为评价指标；最后，对实验结果进行展示和分析，说明了本文所提出的 RACN-SQL 模型和其他的对比模型的参数设置，展示了不同模型在 CSpider 数据集上的准确率、RACN-SQL 模型和 RAT-SQL(Aug)模型 SQL 各成分预测数据以及去除 RACN-SQL 模型中不同模块后的准确率，验证了 RACN-SQL 模型在解决中文复杂 Text-to-SQL 任务的有效性，分析其他模型可能出现的问题，并指出了 Text-to-SQL 任务的重难点和未来研究需要关注的方向。

5 总结与展望

5.1 本文工作总结

近年来，在自然语言处理领域，越来越多以结合工程实践、更好地服务于人们生产生活为目的的研究工作开展。随着深度学习模型在自然语言处理任务上的准确率提高，许多研究工作能落地到实践中，甚至发挥其商业价值，得到了人们的广泛关注。

本课题所研究的将自然语言问题生成为 SQL 语句（Text-to-SQL）任务就属于目前自然语言处理领域的热门研究之一，语义解析任务。Text-to-SQL 任务的目标是根据用户提出的自然语言问题来生成用于数据查询检索的 SQL 语句，交由数据库执行引擎执行，返回给用户所需要的查询结果。Text-to-SQL 任务能够提高数据库的检索效率，降低用户的学习和使用门槛。

纵观国内外的研究现状，早期的研究是针对特定领域的数据库，通过人工制定规则的方式来完成的，需要大量维护成本的同时也不具备通用性。近年来，基于深度学习模型的语义解析研究陆续开展，但这些研究集中在英文数据集上，并且在研究过程中对 SQL 语句的形式有大量限制，虽然简化了模型，但不具备实用性。典型的代表就是在 WikiSQL 数据集上的一系列研究，所有的 SQL 语句都只有 SELECT、FROM、WHERE 三部分，无法满足工程实践中的需要。在中文 Text-to-SQL 领域，由于中文产生歧义几率高、中文自然语言问题和英文数据库模式之间存在跨语言障碍等特点，现有的研究工作甚少。

在分析了国内外研究的不足后，针对中文复杂 Text-to-SQL 任务的特点，本文提出中文自然语言生成复杂 SQL 的模型 RACN-SQL。RACN-SQL 模型将中文复杂 Text-to-SQL 任务分为 5 个关键步骤来解决：（1）自然语言问题与数据库模式的语义编码；（2）数据库模式的关系表示；（3）自然语言问题与数据库模式的关系链接；（4）关系编码；（5）SQL 语句解码。通过这 5 个关键步骤，RACN-SQL 能够生成具有复杂结构的 SQL 语句，也能解决自然语言问题和数据库模式之间跨语言的障碍。

为了验证本文提出的 RACN-SQL 模型的有效性，本文选取了 2019 年发布的中文复杂 SQL 数据集 CSpider 进行实验，该数据集涵盖了不同领域的数据库、不同结构的 SQL 语句，也符合工程实践的特点，即自然语言问题为中文，数据库模式的命名为英文且以下划线作为分词符。实验比较了本文提出的 RACN-SQL 模型与其他在中英文数据集上表现优秀的模型，论证了对于中文复杂 Text-to-SQL 任务，RACN-SQL 模型能通过语义、关系、常识等信息，结合注意力机制和语法树解码，解决跨语言障碍，以较高准确率生成具有复杂结构的 SQL 语句。

与现有的研究相比，本文的主要工作如下：

（1）本文设计了面向 Text-to-SQL 任务的中文自然语言问题和英文数据库模式的跨语言语义编码表示方法。结合不同数据列的类型信息，捕捉用户提出的问题与数据库模式之间的对齐关系、隐含的指代关系。

（2）本文通过有向图的方式将数据库模式显式定义的关系和自然语言问题与数据库模式间的链接关系表示出来。对于数据库模式显式定义的关系，有向图的顶点为数据表或数据列，关系边的类型有主键、外键、包含等。对于自然语言问题与数据库模式间的链接关系，有向图的顶点为自然语言问题中的词和数据表、数据列、数据值，本文将其分为数据表和数据列的链接、数据值的链接两大类，前者在链接时利用 n-gram 统计语言模型对分词后的各个数据表名和数据列名进行链接，用到 ConceptNet 常识语义网络中的同义、相关等关系边，来帮助模型解决跨语言的问题并通过常识来增强关系判断；后者在链接时分为直接在数据库有关数据列中检索数据值、利用 ConceptNet 的相关关系边两种方式进行链接。在链接过程中给不同途径链接得到的关系分配不同的关系标签，在训练中给不同途径获得的信息不同的重要程度权重。

（3）本文提出了中文自然语言生成复杂 SQL 的模型 RACN-SQL，模型细化了 Text-to-SQL 任务的流程，在进行语义编码和关系表示的基础上，基于 Transformer 的多头注意力机制进行关系编码，编码过程中对有向图中的不同关系标签训练不同的注意力偏置权重，得到自然语言问题、数据库模式之间的关系向量输出。

（4）本文通过语法树的解码方法，按照深度优先顺序生成 SQL 语句，将解码

过程分为生成语法关键字和生成数据库模式两类，根据当前结点类型、父结点类型来决定可能出现的候选项，再经过 LSTM 计算概率分布，选取概率值最高的候选项。这种解码方式符合 SQL 语句结构和结构化数据的特点，能保证语法的正确性，支持生成包括表连接、嵌套查询在内的复杂结构的 SQL。

5.2 未来工作展望

本文提出的 RACN-SQL 模型在中文复杂 Text-to-SQL 任务上还有如下可以拓展和完善的方向：

（1）在自然语言问题和数据库模式的语义编码阶段，本文使用 BERT 作为预训练语言模型进行编码。BERT 是自然语言处理中序列到序列（seq2seq）任务的预训练模型，在处理机器翻译、人机问答、文本生成等非结构化数据的任务时表现很好，但在 Text-to-SQL 任务中同时处理非结构化数据的自然语言问题与结构化数据的数据库模式时表现欠佳，目前已有一些前沿研究开始对英文的非结构化数据和结构化数据进行联合训练，得到更加适用于 Text-to-SQL 任务的预训练模型。因此，未来可以关注中文上的联合预训练方法，取代 BERT，进一步完善语义编码过程。

（2）在进行自然语言问题与数据库模式链接时，需要考虑自然语言问题中模糊性的关系，尽可能明确自然语言问题中每个词的实际含义后进行更多精准的链接。在实际的生产实践中，作为用户提出的自然语言问题往往不能直接与数据库模式中的名称相匹配，存在简称、缩写、代词等一系列计算机难以理解的词，导致链接关系减少，进而在生成 SQL 语句时忽略了某些数据库模式。另一方面，在庞大的数据库系统中常存在大量相同的数据值（特别是数字）、数据列名，在进行链接时，需要甄别这些信息之间的关系，当有向图中出现大量关系边噪声时，也会造成最终模型的出错。因此，明确用户提问中词以及数据库模式的含义后进行准确率更高的链接是未来工作的重要方向。

（3）在进行关系编码过程中，本文基于 Transformer，使用了带偏置的多头注意力机制，而不同的偏置向量由不同的关系标签决定。在实际中，相同的标签也可能有不同的含义，比如“同义”关系边连接的两个顶点可能语气不同或者存在正式与

非正式的区别，统一按照“同义”来进行训练可能忽略了这些信息。另一方面，不同的标签之间也可能有相同的含义，无论在中文和英文中都存在大量相近含义的词，在数据库模式的名称中存在大量名称不同但含义相同的列，按照不同的标签训练，会产生噪音，降低准确率。因此，对于关系编码中如何综合考虑非结构化数据和结构化数据之间的关系，减弱用户提问中噪声的影响，针对不同方案开展实验和论证也是未来工作的重要方向。

致 谢

转眼间，我的研究生生涯就要结束了，在两年的科研生活中，经历了许多酸甜苦辣，这些宝贵的经历对我产生了巨大影响，从中我也受益良多。两年间，我不仅学到了大量专业知识、参与了项目的工程实践，也学会了人际沟通的技巧、提高了为人处世的能力。在此我要感谢华中科技大学智能与分布计算（IDC）实验室、家人、老师、同学在我科研之路上给予我的帮助。

首先，我要感谢我的导师辜希武老师，辜老师经验丰富、平易近人，他的课程深受大家喜爱。在科研上，辜老师不辞辛苦地给予我细致的指导，了解我的研究想法和难点之后，提供了很多深入研究的思路，给我指明了方向；在生活上，辜老师经常关注同学们的压力情况，秋招季还会关心我们找工作的进度，提供建议。无论是科研还是生活，辜老师的指导都对我大有裨益。

我还要感谢实验室的主任李瑞轩老师，他学识渊博、认真负责。每周的课题研讨会上都有他的身影，对我们的研究进展的讨论中，李老师总能一针见血指出要点，打开我们的思路，引发我们更深入的思考。虽然李老师工作繁忙，但他总是挤出时间给予同学们指导、也关心同学们的身心健康，实验室每周进行羽毛球活动，鼓励同学们锻炼身体，积极运动。

我也要感谢实验室的李玉华老师，作为我保研时联系的老师，她治学严谨、为人友善，帮助我踏入了科研之门。从和李玉华老师的沟通交流中，我明确了研究生阶段要学习、实践的内容，制定了合理的学习和科研的计划，让我研究生阶段起步很快。

同时，我要感谢王俊博士，在每周的研讨会时，在美国工作的王俊博士都会准时与我们进行线上讨论，给我们的研究工作提出宝贵的指导意见。王俊博士的指导让我们拓宽眼界，了解前沿的研究点，让我们少走了很多弯路。

感谢谢修远、吴小建、崔玉展、李相臣、高鑫等 IDC 实验室中的同学们。两年间大家互相帮助，互相鼓励，一起讨论解决科研工作上遇到的困难，一起分享找工

作过程中的点点滴滴，也祝愿各位同学有美好的前程。

我还要特别感谢我的家人，在读研期间一直给予我鼓励，在背后默默支持着我，在迷茫时给我指出明确的方向，在困难时也会给我宝贵的意见。

最后，感谢各位答辩委员会的老师们对我的论文提出宝贵的审阅指导意见！

参考文献

[1] Pennington J, Socher R, Manning C D. Glove: Global vectors for word representation[C]. Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014: 1532-1543

[2] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality[C]. Advances in neural information processing systems. 2013: 3111-3119

[3] Schuster M, Paliwal K K. Bidirectional recurrent neural networks[J]. IEEE transactions on Signal Processing, 1997, 45(11): 2673-2681

[4] Marmanis D, Datcu M, Esch T, et al. Deep learning earth observation classification using ImageNet pretrained networks[J]. IEEE Geoscience and Remote Sensing Letters, 2015, 13(1): 105-109

[5] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]. Advances in neural information processing systems. 2012: 1097-1105

[6] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780

[7] Zhou P, Qi Z, Zheng S, et al. Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling[J]. arXiv preprint arXiv:1611.06639, 2016

[8] Sundermeyer M, Alkhouli T, Wuebker J, et al. Translation modeling with bidirectional recurrent neural networks[C]. Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014: 14-25

[9] Peters M E, Neumann M, Iyyer M, et al. Deep contextualized word representations[J]. arXiv preprint arXiv:1802.05365, 2018

[10] Radford A, Narasimhan K, Salimans T, et al. Improving language understanding by generative pre-training[J]. 2018

[11] 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

[12] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]. Advances in neural information processing systems. 2017: 5998-6008

[13] Androutsopoulos I, Ritchie G D, Thanisch P. Natural language interfaces to databases-an introduction[J]. arXiv preprint cmp-lg/9503016, 1995

[14] Woods W. The lunar sciences natural language information system[J]. BBN report, 1972

[15] Codd E F, Arnold R S, Cadiou J M, et al. Rendezvous version 1: An experimental English-language query formulation system for casual users of relational data bases[M]. International Business Machines Corporation, 1978

[16] Hendrix G G, Lewis W H. Transportable natural-language interfaces to databases[R]. SRI INTERNATIONAL MENLO PARK CA ARTIFICIAL INTELLIGENCE CENTER, 1981

[17] Warren D H D, Pereira F C N. An efficient easily adaptable system for interpreting natural language queries[J]. American journal of computational linguistics, 1982, 8(3-4): 110-122

[18] 张亚南, 徐洁磐. 数据库 NL 界面上汉语查询的 EAAD 模型[J]. 计算机学报, 1993, 16(12): 881-888

[19] Meng X, Wang S. Nchiql: The chinese natural language interface to databases[C]. International Conference on Database and Expert Systems Applications. Springer, Berlin, Heidelberg, 2001: 145-154

[20] Miller G A, Beckwith R, Fellbaum C, et al. Introduction to WordNet: An on-line lexical database[J]. International journal of lexicography, 1990, 3(4): 235-244

[21] Popescu A M, Etzioni O, Kautz H. Towards a theory of natural language interfaces to databases[C]. Proceedings of the 8th international conference on Intelligent user interfaces. 2003: 149-157

[22] Dong L, Lapata M. Language to logical form with neural attention[J]. arXiv preprint arXiv:1601.01280, 2016

[23] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[C]. Advances in neural information processing systems. 2014: 3104-3112

[24] Zhong V, Xiong C, Socher R. Seq2sql: Generating structured queries from natural

language using reinforcement learning[J]. arXiv preprint arXiv:1709.00103, 2017

[25] Sutton R S, Barto A G. Reinforcement learning: An introduction[M]. MIT press, 2018

[26] Kaelbling L P, Littman M L, Moore A W. Reinforcement learning: A survey[J]. Journal of artificial intelligence research, 1996, 4: 237-285

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

[28] 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

[29] Shi T, Tatwawadi K, Chakrabarti K, et al. Incsql: Training incremental text-to-sql parsers with non-deterministic oracles[J]. arXiv preprint arXiv:1809.05054, 2018

[30] 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

[31] Zhou Z H. A brief introduction to weakly supervised learning[J]. National Science Review, 2018, 5(1): 44-53

[32] Liang C, Norouzi M, Berant J, et al. Memory augmented policy optimization for program synthesis and semantic parsing[C]. Advances in Neural Information Processing Systems. 2018: 9994-10006

[33] Agarwal R, Liang C, Schuurmans D, et al. Learning to generalize from sparse and underspecified rewards[J]. arXiv preprint arXiv:1902.07198, 2019

[34] Yu T, Zhang R, Yang K, et al. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task[J]. arXiv preprint arXiv:1809.08887, 2018

[35] Bogin B, Gardner M, Berant J. Representing schema structure with graph neural networks for text-to-sql parsing[J]. arXiv preprint arXiv:1905.06241, 2019

[36] Guo J, Zhan Z, Gao Y, et al. Towards complex text-to-sql in cross-domain database with intermediate representation[J]. arXiv preprint arXiv:1905.08205, 2019

[37] Zhang R, Yu T, Er H Y, et al. Editing-based sql query generation for cross-domain context-dependent questions[J]. arXiv preprint arXiv:1909.00786, 2019

[38] Choi D H, Shin M C, Kim E G, et al. Ryansql: Recursively applying sketch-based slot fillings for complex text-to-sql in cross-domain databases[J]. arXiv preprint

arXiv:2004.03125, 2020

[39] Wang B, Shin R, Liu X, et al. Rat-sql: Relation-aware schema encoding and linking for text-to-sql parsers[J]. arXiv preprint arXiv:1911.04942, 2019

[40] Shaw P, Uszkoreit J, Vaswani A. Self-attention with relative position representations[J]. arXiv preprint arXiv:1803.02155, 2018

[41] Li Y, Chen B, Liu Q, et al. " What Do You Mean by That?" A Parser-Independent Interactive Approach for Enhancing Text-to-SQL[J]. arXiv preprint arXiv:2011.04151, 2020

[42] Yao Z, Tang Y, Yih W, et al. An imitation game for learning semantic parsers from user interaction[J]. arXiv preprint arXiv:2005.00689, 2020

[43] Zhao L, Cao H, Zhao Y. GP: Context-free Grammar Pre-training for Text-to-SQL Parsers[J]. arXiv preprint arXiv:2101.09901, 2021

[44] Shi P, Ng P, Wang Z, et al. Learning Contextual Representations for Semantic Parsing with Generation-Augmented Pre-Training[J]. arXiv preprint arXiv:2012.10309, 2020

[45] Min Q, Shi Y, Zhang Y. A pilot study for chinese sql semantic parsing[J]. arXiv preprint arXiv:1909.13293, 2019

[46] Yu T, Yasunaga M, Yang K, et al. Syntaxsqlnet: Syntax tree networks for complex and cross-domaintext-to-sql task[J]. arXiv preprint arXiv:1810.05237, 2018

[47] Speer R, Chin J, Havasi C. Conceptnet 5.5: An open multilingual graph of general knowledge[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2017, 31(1)

[48] Ba J L, Kiros J R, Hinton G E. Layer normalization[J]. arXiv preprint arXiv:1607.06450, 2016

[49] Cavnar W B, Trenkle J M. N-gram-based text categorization[C]. Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval. 1994, 161175

附录 1　攻读硕士学位期间参与的项目及取得的学术成果

[1] 横向合作研究项目，基于人工智能技术的金融服务系统开发与应用，2019.6-2021.2

[2] 李瑞轩, **林毅炜**, 辜希武, 李玉华. 一种基于深度学习的中文自然语言生成 SQL 语句方法（发明专利），申请时间：2021.05.21