计算机科学
COMPUTER SCIENCE

图神经网络在 Text-to-SQL 解析中的技术研究

曹合心，赵亮，李雪峰

引用本文

曹合心，赵亮，李雪峰. 图神经网络在 Text-to-SQL 解析中的技术研究[J]. 计算机科学, 2022, 49(4): 110-115.

CAO He-xin, ZHAO Liang, LI Xue-feng. Technical Research of Graph Neural Network for Text-to-SQL Parsing[J]. Computer Science, 2022, 49(4): 110-115.

相似文章推荐（请使用火狐或 IE 浏览器查看文章）

Similar articles recommended (Please use Firefox or IE to view the article)

基于时空自适应图卷积神经网络的脑电信号情绪识别

EEG Emotion Recognition Based on Spatiotemporal Self-Adaptive Graph ConvolutionalNeural Network

计算机科学, 2022, 49(4): 30-36. https://doi.org/10.11896/jsjkx.210900200

融合快速注意力机制的节点无特征网络链路预测算法

Link Prediction for Node Featureless Networks Based on Faster Attention Mechanism

计算机科学, 2022, 49(4): 43-48. https://doi.org/10.11896/jsjkx.210800276

大数据驱动的社会经济地位分析研究综述

Big Data-driven Based Socioeconomic Status Analysis:A Survey

计算机科学, 2022, 49(4): 80-87. https://doi.org/10.11896/jsjkx.211100014

结合绘画先验的线稿上色方法

Sketch Colorization Method with Drawing Prior

计算机科学, 2022, 49(4): 195-202. https://doi.org/10.11896/jsjkx.210300140

一种基于深度学习的供热策略优化方法

Heating Strategy Optimization Method Based on Deep Learning

计算机科学, 2022, 49(4): 263-268. https://doi.org/10.11896/jsjkx.210300155

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.210200173

图神经网络在 Text-to-SQL 解析中的技术研究

曹合心[1]　赵　亮[2]　李雪峰[1]
1 同济大学电子与信息工程学院 上海 201804
2 华东理工大学信息科学与工程学院 上海 200237
(caohexin1994@alumni.tongji.edu.cn)

摘　要　语义解析领域中的 Text-to-SQL 任务对实现基于数据库的自动问答具有重要意义。现有深度学习模型,如 Seq2Seq 的序列生成模型在单表 SQL 查询中已取得显著效果,但无法解决多表 SQL 查询的问题。图神经网络能够有效提取数据库表和问句之间的关联信息,丰富解析过程中的语义信息,从而提升多表 SQL 查询的准确率。文中提出一种自适应的图构建方式和图编码方式,在现有 Text-to-SQL 模型中引入问句信息,通过对问句和数据库的拼接词向量进行卷积操作生成图网络初始化权重,对同种类型下的不同数据库可实现统一训练。采用 IRNet 框架和关系扩充的方式进行整体模型设计,在当前开放的 Text-to-SQL 数据集 Spider 上进行验证。结果表明,该技术能够有效提升多表 SQL 语句生成的匹配准确率,同时算法对图神经网络在 Text-to-SQL 领域的研究具有重要的参考价值。

关键词:Text-to-SQL 解析;深度学习;图构建;图神经网络;多表 SQL 语句生成
中图法分类号　TP391.1

Technical Research of Graph Neural Network for Text-to-SQL Parsing

CAO He-xin[1], ZHAO Liang[2] and LI Xue-feng[1]
1 College of Electronic and Information Engineering, Tongji University, Shanghai 201804, China
2 College of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, China

Abstract　The Text-to-SQL task in the field of semantic parsing is of great significance for realizing database-based automatic question and answer. At present, deep learning models, such as sequence generation model Seq2Seq, has achieved significant effects in single-table SQL queries. However, the problem of multi-table SQL queries remains to be solved. Graph neural network can effectively extract the associated information between databases, tables and questions, enrich the semantic information in the parsing process, and improve the accuracy of multi-table SQL queries. This paper proposes an adaptive graph construction method and graph encoding method. Question information is introduced into the existing Text-to-SQL model, and the graph network initialized weights are generated by convolution operation on the splicing word vector of the question sentence and the database. General training can be achieved for different databases of the same type. The IRNet framework and relational expansion are used to design the overall model, and it is verified on the open Text-to-SQL data set——Spider. Results show that the technology can effectively improve the matching accuracy of multi-table SQL statement generation, and the algorithm has an important reference value for the research of graph neural network in the text-to-SQL field.

Keywords　Text-to-SQL parsing, Deep learning, Graph construction, Graph neural network, Multi-table SQL statement generation

1　引言

随着计算机水平的快速发展,数据的获取更加便捷,数据的搜索和查询对于企业生产和人民的日常生活具有重要的导向作用。现有数据通常存储在关系型数据库系统,而研究人员获取所需数据需要掌握数据库相关知识以及基于 SQL (Structured Query Language)的查询语言,其专业技能要求较高,对于一般用户而言难以实现。近年来,随着人工智能技术的发展,通过对自然语言的解析,直接生成和数据库系统交互的 SQL 语句已成为研究热点之一。目前的研究工作通常采用一个自然语言-数据库接口(Natural Language Interface to Database, NLIDB),来实现用户的问题与数据库系统的交互,从而获取和分析数据[1]。

NLIDB 的核心问题就是将输入的文本信息转化为 SQL

到稿日期:2021-02-26　返修日期:2021-07-07
基金项目:中央高校基本科研业务费专项资金(2212015665)
This work was supported by the Fundamental Research Funds for the Central Universities of Ministry of Education of China(2212015665).
通信作者:李雪峰(lixuefeng@tongji.edu.cn)

语句(Text-to-SQL)。为了解决该问题，目前可分为两个研究方向：1)基于规则模板的方式，即按照常用 SQL 语法对自然语言进行分类，并对不同类别配以对应的 SQL 模板。此类方法需要人工总结经验，具有较高的时间成本。此外，随着应用场景的切换，现有模板常常难以满足要求，迁移性较差。2)基于深度学习的方式，采用神经网络进行端到端的实现。该方法只需通过不断加入样本信息即可自我优化，具有准确性高、适应性强等优点，正受到学界越来越广泛的关注。

Text-to-SQL 的深度学习模型需要结合数据库本身的信息和用户的自然语言，提取出关联的数据库表名、列名以及用户的意图模式。然而数据库本身复杂的内部结构给常规的深度学习模型带来了很大挑战，主要体现在：1)传统的深度神经网络的输入为结构化数据，而数据库中存在表名、列名和列名对应的具体数据，整合多种信息源无疑增加了流程的复杂性；2)数据库的各元素之间存在显著依存关系，尤其在多表 SQL 查询场景中，有效提取表-表、表-主键、主键-外键等关联信息并代入模型，从而生成 SQL 的 join 部分语句(见图 1)，也是一个主要难题。近年来，越来越多的深度学习方法引入了图网络这一形式。图神经网络(Graph Neural Network，GNN)通过将源数据转化为关系图谱作为输入，采用聚集邻近节点的方式，在图结构上进行线性变换、卷积等操作[2]，能够准确表达数据库各元素的关联信息，可以解决传统深度学习模型遇到的上述问题。

用户问题：For each stadium，how many concerts play there?

SQL：SELECT T2. name，count(*) FROM concert AS T1 JOIN stadium AS T2 ON T1. stadium_id= T2. stadium_id GROUP BY T1. stadium_id

图 1　多表连接的复杂 SQL 语句

Fig. 1 Complex SQL statement for multi-table connection

本文在 Text-to-SQL 领域提出了一种解决办法。本文的主要工作如下：

(1)提出了一种基于用户问句和数据库信息的图网络自动构建方式，并从中总结出 39 种关系类型信息，有助于提升识别问句中关键词的识别准确率。

(2)设计了一种自适应的图自动编码技术，与先前的研究相比，其具有跨数据库的通用性优点。

(3)在开放的 Text-to-SQL 任务数据集 Spider 上进行验证，本文方法能够有效提升 SQL 的匹配准确率。

2　相关工作

自然语言接口目前已在工业领域产生诸多应用，如聊天机器人、自动问答系统等。近几十年来，基于自然语言的数据查询技术仍然是数据库和自然语言处理领域的热点难题。研究人员在早期通常采用人工的方式总结规则模板[3]。2004 年，Popesu 等将自然语言的子集定义为语义上易处理的查询[4]，并且精确地将这些查询转换为相应的 SQL 查询，但对于语义上难以处理的问句则无法进行匹配。2012 年，Unger 等基于 SPARQL 语法将用户的问题转换为一种中间模板[5]，并采用 WordNet 实现模板中具体要素的预测。2014 年，Li 等[6]设计出一种基于“解析树”进行中间表达的 NaLIR 系统。该系统需要用户主动选择某种树结构，然后通过解码器将树图翻译为 SQL 语句。此类方法在单独场景中的准确率较高，但需要人力参与，且只能针对特定的数据库单独定制，不具有通用性。

深度学习的发展使得基于神经网络的端到端流程得以实现。2017 年 Zhong 等利用增强学习的相关知识，提出基于策略模式的 Seq2SQL 模型[7]，在 WikiSQL 数据集上单表查询的 SQL 执行准确率达到 59.4%。同年，Yu 等提出 TypeSQL 模型，通过结合外部知识和数据库字段枚举值对问句中的关键词进行进一步抽取[8]。以上方法在单表查询中取得了明显效果，但不足以解决多表查询的复杂模式。2018 年，Yu 等采用融合 SQL 生成历史记录和数据库信息的编码器以及基于 SQL 特定语法树的解码器，提出一种 SyntaxSQL 模型[9]，对于具有多表查询的 Spider 数据集[10]，其查询准确率在原有模型的基础上提升了 14.8%。2020 年，Cao 等利用斯坦纳树理论解决 From 模块的多表 join 路径不匹配问题[11]。同年，Zhang 等采用编辑机制引入用户查询的历史信息，提出 EditSQL 框架[12]，在 spider 数据的测试集上准确率提升至 32.9%。

传统神经网络模型难以处理非结构化数据，对于多表 SQL 解析问题，如何识别问句中的字段所在表，及表-表、表-字段和字段-字段的连接关系，对提升模型的效果具有重要作用。图神经网络可以读取数据库层级的诸多元素关联信息，采用邻居节点聚合的方式将数据库信息进行编码，从而有助于识别问句中的列名和表名。2019 年，Bogin 等使用 Gated GNN 对数据库模式的结构进行训练，并在编码和解码阶段同时使用该形式[13]，准确率提高至 39.4%，此后，通过添加全局节点的方式引入 Global GNN 模型，进一步将准确率提升至 47.4%[14]。2020 年，Xie 等提出一种将 SQL 查询语句变化成图网络的方法，为后续研究提供了参考方向[15]。然而在 Text-to-SQL 领域中，现有的图构建方式仅仅考虑了数据库层面的关联信息，尚未将问句的词向量作为本体融入图网络。此外，由于不同数据库具有不同的关联关系，即使对于同一种存储类型的数据库，其也需要分别在每个数据库中训练 GNN 模型，不具有迁移性。而本文提出的架构有助于解决该类问题。

3　任务定义

为了准确描述 Text-to-SQL 任务需要研究的相关问题，本节给出了所设计概念的具体定义。

3.1　模型输入

Text-to-SQL 任务模型的输入通常由两部分组成：

(1)问题集合 Q：

$$Q=\{q_1,q_2,\cdots,q_k,\cdots q_n\} \tag{1}$$

其中，q_k 为问句 Q 中的第 k 个词语，n 为问句的词向量个数。

(2)数据库集合 S，包括表名集合 T 和列名集合 C。

$$T=\{t_1,t_2,\cdots,t_k,\cdots,t_m\} \tag{2}$$

$$C=\{c_{t11},c_{12},\cdots,c_{kj},\cdots,c_{ml}\} \tag{3}$$

其中，t_k 为表集合 T 中的第 k 张表，m 为数据库中的总表数，c_{kj} 为第 k 张表的第 j 个列名，c_{ml} 为最后一张表的最后一个列名。

此外，本文引入第三部分关系集合 R 作为图神经网络的输入信息：

$$R=\{r_1,r_2,\cdots,r_k,\cdots,r_{39}\} \tag{4}$$

考虑问句词和数据库元素之间的诸多对应关系，本文总结出 39 种关系类型。

3.2 模型输出

本文的模型解析部分采用中间表示层框架[15]，即按照 SQL 语法抽象出若干独立元素，模型输出元素序列即 SemQL 语句。SemQL 由 sketch 和 A/C/T 两部分组成。前者由 Root1，Root，Sel，Order，Sup 和 Filter 模块组成，每个模块对应多种子模式。A/C/T 分别表示运算符、列名索引和表名的索引。通过对 SemQL 的各模式进行编号，可将其映射成最终的 SQL 语言。由 SQL 拆分成的 SemQL 各个元素如图 2 所示。

```
Root1→Root+intersect|union|except|none (+Root)
Root→Sel|Sel Filter|Sel Order|Sel Sup|Sel Order Filter|Sel Sup Filter
Sel→A|A A|A A A|A A A A|A A…A
Order→asc A|desc A          Sup →most A|least A
Filter→and Filter Filter|or Filter Filter
        |>A|>A Root|<A|<A Root
        |>=A|>=A Root|=A|=A Root
        |!=A|!=A Root|between A
        |like A|not like A|in A Root|not in A Root
A→max C T|min C T|count C T
    | sum C T|avg C T|none C T
C→column                    T→table
例：
SQL:ELECT song_name,song_release_year FROM singer. ORDER BY age LIMIT 1
SemQL:Root1(3) Root(2) Sel(0) A(0) C(11) T(1) A(0) C(12) T(1) Sup(1) A(0) C(13) T(1)
```

图 2 SemQL 语法示意图

Fig. 2 SemQL syntax diagram

4 模型与算法

4.1 模型设计

本文模型整体部分基于 IRNet 的架构[16]，如图 3 所示。底层输入采用问句和数据库列名表名拼接的方式，通过自然语言预训练模型 Bert 的 transformer 结构[17]输出各自的向量 $\boldsymbol{Wq}_0$，$\boldsymbol{Wc}_0$，$\boldsymbol{Wt}_0$，其中对列名和表名的多个词取平均值，作为最终的向量表示。随后 $\boldsymbol{Wq}_0$ 经过全连阶层转化为 300 维的问句词向量 $\boldsymbol{Wq}$，$\boldsymbol{Wc}_0$ 和 $\boldsymbol{Wt}_0$ 各自通过一个双向长短时记忆网络(Long Short-Term Memory，LSTM)，同样形成 300 维的列名向量 $\boldsymbol{Wc}$ 和表名向量 $\boldsymbol{Wt}$，并与经过本文设计的 GNN 编码后的输出向量进行叠加。其中问句部分采用 Attention 机制结合 LSTM 输出 SemQL 的 sketch 部分。列名和表名解析则分别采用不同的指针网络模型(Pointer Network)，并引入问句的输出向量预测运算符 A 的所属类别，最后将二者结合通过规则解析构成 SQL 语言。

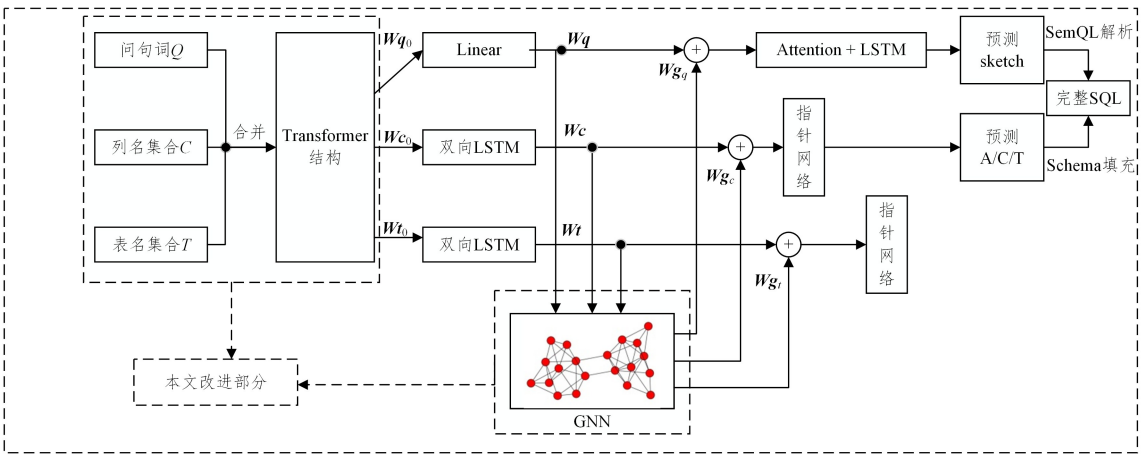

图 3 本文模型的整体架构

Fig. 3 Overall architecture of model in this paper

4.2 图网络构建

现有 Text-to-SQL 任务采用数据库信息构建图网络，仅仅含数据库表名-表名、表名-列名和列名-列名 3 种主要关系类型。本文采用了一种区别于以往 GNN 图谱的构建方式，即在原有方法上引入问句信息，新增了问句词-问句词、问句词-表名、问句词-列名这 3 类主要关系，并在原有关系的基础上进行细化和扩充。

本文所构建图网络包含 3 类实体节点：问句词、列名和表名。考虑到问句词之间的距离及其与列名和表名的匹配方式，形成了共计 39 种关系类型的全连接关系图谱。

(1)列名—>列名($c_i \rightarrow c_j$)：相同列名(r_0)，c_i 是 c_j 外键(r_1)，c_j 是 c_i 外键(r_2)，属于同一张表(r_3)，不属于同一张表(r_4)。

(2)列名—>表名($c_i \rightarrow t_j$)：c_i是t_j主键(r_5)，c_i是t_j的其中一列(r_6)，c_i不属于t_j的其中一列(r_7)。

(3)表名—>列名($t_i \rightarrow c_j$)：t_i存在主键c_j(r_8)，t_i存在列c_j(r_9)，t_i不存在列c_j(r_{10})。

(4)表名—>表名($t_i \rightarrow t_j$)：相同表名(r_{11})，t_i通过外键连接t_j(r_{12})，t_j通过外键连接t_i(r_{13})，两表互为外键连接形式(r_{14})，两表不存在直接连接(r_{15})。

(5)列名—>问句词($c_i \rightarrow q_j$)：c_i完全匹配q_j(r_{16})，c_i部分匹配q_j(r_{17})，二者不存在连接关系(r_{18})。

(6)问句词—>列名($q_i \rightarrow c_j$)：q_i完全匹配c_j(r_{19})，q_i部分匹配c_j(r_{20})，二者不存在连接关系(r_{21})。

(7)表名—>问句词($t_i \rightarrow q_j$)：t_i完全匹配q_j(r_{22})，t_i部分匹配q_j(r_{23})，二者不存在连接关系(r_{24})。

(8)问句词—>表名($q_i \rightarrow t_j$)：q_i完全匹配t_j(r_{25})，q_i部分匹配t_j(r_{26})，二者不存在连接关系(r_{27})。

(9)问句词—>问句词($q_i \rightarrow q_j$)：计算问句词q_i到q_j的相对距离d，按照d的大小分别为0～5或大于5，共有r_{28}～r_{34}这种关系。

(10)数据库—>数据库($s_i \rightarrow s_j$)：与数据库层级关系r_0～r_{15}混合计算。如果关联节点均不含问句词，则按照列名-列名(r_{35})、表名-表名(r_{36})、表名-列名(r_{37})、表名-表(r_{38})名划分成4种关系。

其中，对于完全匹配，问句词即是列名或表名，部分匹配则指问句词属于某个列名或表名的一部分。

4.3 基于自编码器的图神经网络

图自编码器(Graph Auto-Encoder，GAE)采用encoder-decoder结构，可以对关系网络进行结构化感知，其生成的节点embedding能够支持下游预测任务。2016年，Kipf等设计出一种变分图自编码器[18]，在encoder部分借用GNN模型通过隐变量学习到节点的特征分布情况，并从中进行采样得到节点embedding，并在decoder部分进行内积运算重构图形，但该方法的隐变量只能针对特定图谱。本文设计出一种新型的GNN算法，对于Text-to-SQL任务可实现由问句词、列名和表名的多结构网络统一编码和训练，同时能够考虑多种关系类型信息。GNN架构的详细推导如下。

(1)将问句词和数据库元素作为实体构建图网络G，由图4可知，将$\boldsymbol{Wq}$，$\boldsymbol{Wc}$，$\boldsymbol{Wt}$通过纵向拼接和补零形成维度为512×300的向量$\boldsymbol{W}_{\text{cat1}}$作为$G$的节点特征矩阵，同时初始化矩阵$\boldsymbol{W}_r=[x_1, x_2, \cdots, x_k, \cdots, x_{39}]$作为关系权重矩阵($x_k$表示第$k$种关系的权重)，对于关系网络$G$的邻接矩阵$\boldsymbol{A}_g$而言，其在第$i$行、第$j$列的值$x_k$表示实体$i$和$j$对应关系$r_k$的权重。

(2)将$\boldsymbol{W}_{\text{cat1}}$的转置矩阵$\boldsymbol{W}^t_{\text{cat1}}$通过卷积和全连接层形成300×512的节点权重矩阵$\boldsymbol{Wei}_1$：

$$\boldsymbol{Wei}_1 = linear(conv(\boldsymbol{W}^t_{\text{cat1}})) \tag{5}$$

(3)将特征矩阵$\boldsymbol{W}_{\text{cat}}$的有效部分与$\boldsymbol{Wei}_1$相乘，得到中间矩阵$\boldsymbol{W}_{\text{mid}}$，其与邻接矩阵$\boldsymbol{A}_g$的乘积经过激活函数形成第一层GNN输出结果$\boldsymbol{Gnn_out}_1$：

$$\boldsymbol{W}_{\text{mid1}} = \boldsymbol{W}_{\text{cat1}}[:l]\boldsymbol{Wei}_1 \tag{6}$$

$$\boldsymbol{Gnn_out}_1 = \text{softmax}(\tanh(\boldsymbol{A}_g\boldsymbol{W}_{\text{mid1}})) \tag{7}$$

其中，l表示$\boldsymbol{W}_{\text{cat}}$补零前的词向量长度；

(4)$\boldsymbol{Gnn_out}_1$做补零、转置后变为512×512的向量$\boldsymbol{Gnn_mid}^t_1$，将其作为第二层权重矩阵的输入，这里将$\boldsymbol{Gnn_mid}^t_1$通过步骤(2)的变换分别产生$\boldsymbol{Wei}_2$和$\boldsymbol{Wei}_3$两个相同矩阵：

$$\boldsymbol{Wei}_2 = linear(conv(\boldsymbol{Gnn_mid}^t_1)) \tag{8}$$

$$\boldsymbol{Wei}_3 = linear(conv(\boldsymbol{Gnn_mid}^t_1)) \tag{9}$$

(5)将$\boldsymbol{Gnn_out}_1$视为第二层GNN特征矩阵，将$\boldsymbol{Wei}_2$和$\boldsymbol{Wei}_3$代入式(6)、式(7)，则得到第二层的两部分GNN输出：

$$\boldsymbol{W}_{\text{mid2}} = \boldsymbol{Gnn_out}_1\boldsymbol{Wei}_2 \tag{10}$$

$$\boldsymbol{W}_{\text{mid3}} = \boldsymbol{Gnn_out}_1\boldsymbol{Wei}_3 \tag{11}$$

$$\boldsymbol{Gnn_out}_2 = \text{softmax}(\tanh(\boldsymbol{A}_g\boldsymbol{W}_{\text{mid2}})) \tag{12}$$

$$\boldsymbol{Gnn_out}_3 = \text{softmax}(\tanh(\boldsymbol{A}_g\boldsymbol{W}_{\text{mid3}})) \tag{13}$$

(6)$\boldsymbol{Gnn_out}_3$分别按照指数分布和随机分布生成对应向量，与$\boldsymbol{Gnn_out}_2$合并运算经激活函数层形成GNN部分的encoder向量$\boldsymbol{Wg}$：

$$\boldsymbol{E}_{\text{std}} = exp_like(\boldsymbol{Gnn}_{\text{out 3}}) \tag{14}$$

$$\boldsymbol{E}_{\text{eps}} = randn_like(\boldsymbol{E}_{\text{std}}) \tag{15}$$

$$\boldsymbol{E}_{\text{union}} = \boldsymbol{E}_{\text{eps}}\boldsymbol{E}_{\text{std}} + \boldsymbol{Gnn_out}_2 \tag{16}$$

$$\boldsymbol{Wg} = \text{softmax}(\tanh(\boldsymbol{E}_{\text{union}})) \tag{17}$$

将$\boldsymbol{Wg}$拆分成对应的问句$\boldsymbol{Wg}_q$、列名$\boldsymbol{Wg}_c$和表名$\boldsymbol{Wg}_t$，加入原模型embedding向量：

$$\boldsymbol{Wq} = \boldsymbol{Wq} + \boldsymbol{Wg}_q \tag{18}$$

$$\boldsymbol{Wc} = \boldsymbol{Wc} + \boldsymbol{Wg}_c \tag{19}$$

$$\boldsymbol{Wt} = \boldsymbol{Wt} + \boldsymbol{Wg}_t \tag{20}$$

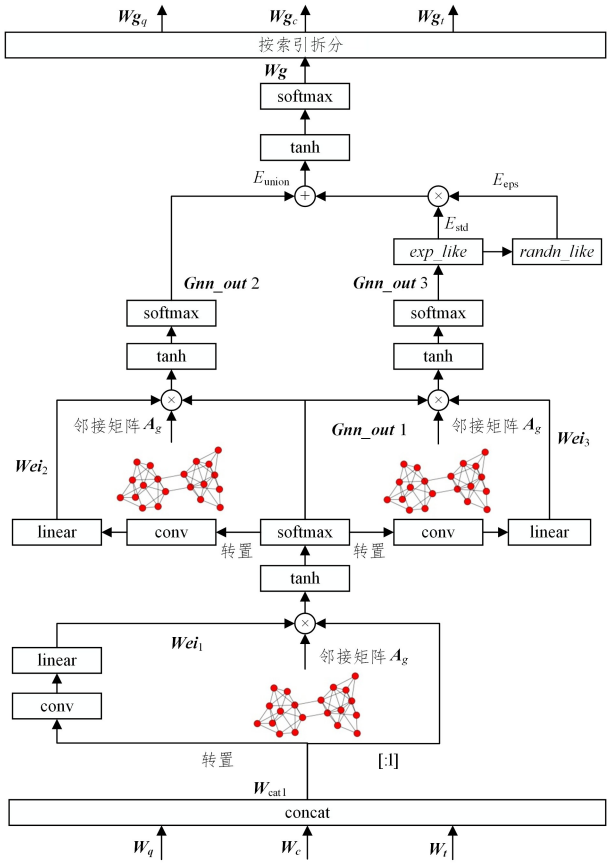

图4 Text-to-SQL任务中的GNN架构

Fig. 4 GNN architecture in Text-to-SQL task

5 实验

本文验证设备为一台Linux服务器，版本为Ubuntu

16.04.4，GPU 为 16 GB 的 NVIDIA Tesla V100。本文基于 Pytorch 的深度学习框架进行模型实现，采用 Python 编程语言，输出 300 维的隐向量，drop-out rate 为 0.3，优化部分使用 Adam，批样本数目为 4，训练时迭代次数为 100，损失函数沿用 IRNet 的计算方式。

5.1 数据集选择

本文使用的数据集为 Spider 数据集，如表 1 所列。该数据集具有丰富的数据库表结构信息，且对于每个自然语言问题都有相应的 SQL 语句进行参考，其优点体现在：

(1)Spider 根据难度分为 Easy，Medium，Hard 和 Extra Hard 共 4 类数据集，是 Text-to-SQL 任务的公开数据集中唯一既包含多表查询情况又存在复杂 SQL 语句的数据集。其中复杂 SQL 指语句中存在 ORDER BY，GROUP BY 以及 WHERE 子句的嵌套查询情况，适用于评价 GNN 对于多表连接信息的表达程度。

(2)自然语言问句中存在字段的具体枚举值信息，更接近真实问答场景，可有效定位至具体字段，有助于提升预测结果。

(3)数据集由训练集、验证集和测试集 3 部分构成，每个部分采用不同的数据库表结构，可衡量模型对于新数据的迁移能力。

表 1 Spider 数据集的规模

Table 1 Size of Spider

	样本数目	数据库数目
训练集	8659	146
验证集	1034	20
测试集	2147	40

5.2 评价指标

本文采用完全匹配准确率(Exact Matching Accuracy)和部分匹配分值(Partial Matching Score)两种方式对实验结果进行评价。其中，完全匹配准确率指采用标准化定义的方式处理预测 SQL 和真实值，从整体上衡量二者间的匹配程度，无须考虑预测列名的顺序问题；部分匹配分值则将 SQL 语句拆分成 SELECT，WHERE，GROUP BY，ORDER BY 等子句，并逐一计算预测值和真实值在各部分的 $F1$ 值。

5.3 实验结果及分析

表 2 列出了不同模型分别在 Spider 验证集(Dev)和测试集(Test)上的匹配准确率情况。

表 2 不同模型分别在验证集和测试集上的完全匹配准确率对比

Table 2 Comparison of exact matching accuracy for different models on development and test sets

(单位：%)

模型	Dev	Test
Seq2SQL	1.9	3.7
TypeSQL	8.0	8.2
SytaxSQL	24.8	27.2
EditSQL	36.4	32.9
Gated GNN	40.7	39.4
Global GNN	52.1	47.4
IRNet	53.2	46.7
本文	60.9	60.1

由表 2 可知，对于 Text-to-SQL 任务，基于中间表示层的 IRNet 较传统的 GNN 模型在性能上有明显提升，而本文新引入的编码器图神经网络，其最终准确率在原有模型的基础上，在验证集上提升了 7.7%，在测试集上提升了 13.4%，具有明显效果。由于本文方法嵌入了多种数据库及问句词的关联关系，在模型训练时提供了更丰富的表征信息[19]，具有更好的泛化效果，其验证集和测试集的查询准确率基本相近，与原 IRNet 模型相比，有效提升了模型的稳定性。

为进一步研究算法在 SQL 不同部分的预测效果，实验分析了模型的主要模块在 Spider 多个难度下的 $F1$ 值，如表 3 所列。可以发现，由于进行编码时嵌入了数据库元素的关联信息，对于预测列名和表名的部分如 SELECT，GROUP BY，ORDER，其 $F1$ 值普遍较高，说明本文设计的图编码方法具有显著的效果。为了证明模型的鲁棒性，观察 $F1$ 值随着数据集难度由简单逐渐增加至困难级别的变化情况，发现其并没有明显降低，可知模型对于不同难度的数据均有良好效果。

表 3 测试集种不同难度下各模块的 $F1$ 值

Table 3 $F1$ scores of test set for different parts by hardness level

(单位：%)

模块	Easy	Medium	Hard	Extra hard
SELECT	0.927	0.794	0.914	0.759
WHERE	0.889	0.808	0.564	0.564
GROUP BY	0.900	0.752	0.769	0.696
ORDER BY	1.000	0.720	0.836	0.823

此外，实验发现，作为影响模型最终准确率的一个较为关键因素，WHERE 部分的 $F1$ 值在困难模式下偏低。经分析可得，WHERE 部分不仅包含列名解析，还需要从问句中识别如">" "<""==”等条件符号。此类信息无法加入图编码器，因此无法实现文中的提升效果。

结束语 本文针对语义解析领域的 Text-to-SQL 任务，设计出一种图谱构建方法，并提出一种可跨数据库训练的图神经网络。基于用户的问句词和数据库信息，在模型输入阶段构建出包含 39 种关系的异质关系图谱，通过图邻接矩阵的方式，将关系信息进行编码，并进行统一训练。实验表明，与之前的模型相比，该方法能够有效提升 Spider 数据集下多表查询场景的匹配准确率。此外，由于补充了数据库和问句间的表征关系，使得验证集和测试集的结果不存在明显差异，一定程度上提升了模型的稳定性和鲁棒性。由于 SQL 语法在 WHERE 部分需要进行条件判断，而本文在构建图谱时并未考虑该部分因素，因此可在未来工作中探索新的关系设计方式。

参 考 文 献

[1] LBAIK C, JAGADISH H V, LI Y Y. Bridging the semantic gap with SQL query logs in natural language interfaces to databases[J]. arXiv:1902.00031, 2019.

[2] WU Z, PAN S, CHEN F, et al. A Comprehensive Survey on Graph Neural Networks[J]. IEEE Transactions on Neural Networks and Learning Systems, 2020(99): 1-21.

[3] ANDROUTSOPOULOS I, RITCHIE G D, THANISCH P. Natural Language Interfaces to Databases—An Introduction[J]. Natural Language Engineering, 1995, 1(1): 29-81.

[4] POPESCU A M, ARMANASU A, ETZIONI O, et al. Modern natural language interfaces to databases: Composing statistical parsing with semantic tractability[C]// COLING. 2004.

[5] UNGER C, BÜHMANN L, LEHMANN J, et al. Template-based question answering over RDF data[C]// Proceedings of the 21st World Wide Web Conference. New York: ACM, 2012: 639-648.

[6] LI F, JAGADISH H V. Constructing an interactive natural language interface for relational databases[J]. Proceedings of the VLDB Endowment, 2014, 8(1): 73-84.

[7] ZHONG V, XIONG C, SOCHER R. Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning[J]. arXiv: 1709. 0010. 3v7.

[8] YU T, LI Z F, ZHANG Z L, et al. TypeSQL: Knowledge-Based Type-Aware Neural Text-to-SQL Generation[J]. arXiv: 1804. 09769v1, 2018.

[9] YU T, MICHIHIRO Y, KAI Y, et al. SyntaxSQLNet: Syntax Tree Networks for Complex and Cross-DomainText-to-SQL Task[J]. arXiv: 1810. 05237v1, 2018.

[10] 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: 1809. 08887v5, 2019.

[11] CAO J C, HUANG T, CHEN G, et al. Research on Natural Language Generating Multi-table SQL Query Statement Technology [J]. Journal of Computer Science and Research, 2020(7): 1133-1141.

[12] ZHANG R, YU T, ER H Y, et al. Editing-Based SQL Query Generation for Cross-Domain Context-Dependent Questions [C]// Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 2019.

[13] BOGIN B, GARDNER M, BERANT J. Representing Schema Structure with Graph Neural Networks for Text-to-SQL Parsing[J]. arXiv: 1905, 06241v2, 2019.

[14] BOGIN B, GARDNER M, BERANT J. Global Reasoning over Database Structures for Text-to-SQL Parsing[J]. arXiv: 1908. 11214v1, 2019.

[15] XIE J X, GAN Y J, YU G J. Design and Implementation of Algorithms for Converting SQL Query Statements into Graph Structures[J]. Electromechanical Information, 2020(17): 120-123.

[16] GUO J, ZHAN Z, GAO Y, et al. Towards Complex Text-to-SQL in Cross-Domain Database with Intermediate Representation [C]// Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019.

[17] ASHISH V, NOAM S, NIKI P, et al. Attention is All You Need[J]. arXiv: 1706. 03762.

[18] KIPF T N, WELLING M. Variational Graph Auto-Encoders [J]. arXiv: 1611. 07308v1.

[19] YUAN Z X, REN D D, HONG X D, et al. Research on Question Understanding Method Combining Database Structure and Content [J]. Computer Engineering, 2021, 47(3): 71-76, 82.

CAO He-xin, born in 1994, postgraduate. His main research interests include natural language processing and know-ledge graph.

LI Xue-feng, born in 1975, Ph.D, associate professor. His main research interests include NDT, pattern recognition algorithm and signal processing.

(责任编辑:柯颖)